


ANO III - Nº 32
MAIO/84
Cr\$ 2.000,00
ISSN 0101 3041

Micro Sistemas

A PRIMEIRA REVISTA BRASILEIRA DE MICROCOMPUTADORES

EDUARDO M.



**Faça de seu micro um detetive
com os programas que raciocinam**

**Apple: a rotina
que aponta erros**

**Batalha Naval
na linha Sinclair**

**Utilize ainda
mais seu DOS 500**

AS-1000

o micro que cresce com você.



CARACTERÍSTICAS TÉCNICAS:

- 16 K bytes de memória iniciais
- Expansão Interna para 32 e 48 K bytes
- 8 K bytes de memória EPROM
- Microprocessador Z-80A
- Teclado de membrana com ação mecânica positiva
- 40 teclas e 154 funções
- Basic e linguagem de máquina
- Video normal ou reverso
- Saída para qualquer impressora
- Manipula até quatro cassetes com geração de arquivo
- Modem
- Joystick
- Speed File
- Fonte de alimentação embutida (110/220 volts)
- Nível de leitura de gravação automático

O Microcomputador AS-1000 é uma ótima escolha para quem está iniciando na ciência da computação. Seus recursos de programação e sua concepção modular, porém, permitem que ele o acompanhe até as aplicações mais sofisticadas.

O AS-1000 já nasce com uma biblioteca de milhares de programas para jogos, administração doméstica, aplicações comerciais e profissionais.

O AS-1000 é fabricado com a qualidade ENGEBRÁS e garantido por um ano.

Entre na era da informática com a escolha certa. AS-1000, o seu micro pessoal.

Escreva-nos, sua correspondência não ficará sem resposta.

E3

ENGEBRÁS

ELETRÔNICA E INFORMÁTICA LTDA.

Rua do Russel, 450 - 3º andar

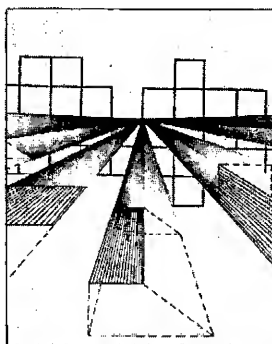
cep 22210 Rio de Janeiro - RJ

Tel.: (021) 205-4898

14 PROGRAMAS QUE RACIOCINAM -

Acompanhe com o prof. Antônio Costa este interessante relato sobre os sistemas especialistas e seu poder de raciocínio lógico e dedutivo. Como exemplo, um Programa Detetive para que seu micro possa demonstrar sua inteligência ao desvendar o mistério de uma investigação policial.

(A CAMAREIRA GASTAVA MUITO)
(A CAMAREIRA BANHAVA POUCO))
(A GERALDA ROUBOU O COLAR)
SE
(A JANELA ESTAVA ABERTA)
(O COLAR ERA BRILHANTE)
(GERALDA ESTAVA DE COISAS BRILHANTES)
(A JANELA ESTAVA ABERTA)
SE
(NÃO ESTAVA CHOVENDO)
(ESTAVA FAZENDO CALOR))
(ESTAVA FAZENDO CALOR)
SE
(ERA VERDADE))
(OS CIGANOS ROUBARAM O COLAR)
SE
(UM CIGANO ENTROU NA CASA)
(UM CIGANO FOI VISTO COM O COLAR)
(A CAMAREIRA BANHAVA POUCO)

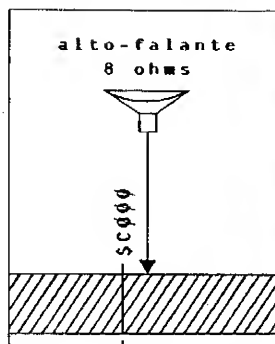


66 EM GEOMETRIA, A ÁREA É DO MICRO -

Arquitetos, engenheiros, matemáticos... você, usuário da linha TRS-80: aproveite os recursos deste programa de Heber Jorge da Silva que desenha e calcula a área de 20 figuras geométricas diferentes, bastando que você entre com as medidas necessárias.

26 SIDRA - Tiros de raio laser! Helicópteros voando! Este mês, Rudolf Horner Jr. ensina como você

pode obter efeitos especiais como estes utilizando recursos do próprio Apple. E mais: como registrar e recuperar informações em fita cassete com mais eficiência e rapidez.



70 BATALHA NAVAL -

Viva as emoções desta verdadeira guerra de nervos e perspicácia contra seu micro da linha Sinclair. Esconda bem os seus, destrua todos os navios de seu inimigo e seja o vencedor deste jogo escrito por Waldir Ferreira de Arruda.

8 A ROTINA APONTA-ERROS -

Programa de Carlos A. Diz.

13 GROAN: OS DADOS ESTÃO EM JOGO -

Programa de José Roberto Côtin.

20 CORRIGINDO PROVAS NO MICRO -

Programa de Lawrence Falconer King.

34 PARA BOM INVESTIMENTO, UM MICRO PESSOAL BASTA -

Programa de Armando Oscar Cavanha Filho.

40 ABRINDO ESPAÇO PARA A LINGUAGEM DE MÁQUINA -

Programa de José Ricardo Flores Rodrigues.

44 DOMINE O MICRO NO DOMINÓ -

Programa de Everton Pereira.

50 DE AZIMUTE EM RUMO -

Programa de Laci Mota Alves.

54 MICRO BUG: O COMANDO M E AS PRINCIPAIS ROTINAS SGM -

Artigo elaborado pela equipe do CPD MS.

62 OUTRAS PALAVRAS EM FORTH -

Artigo de Ivan Camilo da Cruz.

76 MANTENHA SEUS BYTES SOB CONTROLE -

Artigo de Carlos A. Diz.

80 ALÉM DO BASIC, ASSEMBLER, DOS E OUTROS MISTÉRIOS -

Programa de Roberto Quito de Sant'Anna.

84 CURSO DE ASSEMBLER - XVI

SEÇÕES

30 BITS

46 LIVROS

4 EDITORIAL

38 XADREZ

48 DICAS

6 CARTAS

42 CLASSIFICADOS

75 MENSAGEM DE ERRO



editorial

Justo no momento em que o mercado brasileiro começa a adentrar mais claramente na discussão sobre os equipamentos de 16 bits, 32 bits, micrinhos, supermicros e outros bichos, uma verdade inofismável pode ser detectada: a micro-informática no Brasil ainda não constitui uma comunidade de usuários que eficientemente interaja com a máquina. Culpa dos existem por toda parte e não se trata, aqui, de expurgá-los, visto que o remédio neste caso pode acabar matando o paciente, mas é preciso exorcisar alguns fantasmas que insistem em permanecer no cenário.

A profusão de modelos tipo XLMW/33 (o "X" é experimental), compatível apenas consigo mesmo — alterou um endereçozinho do Apple, um parafusinho

no TRS, um pequeno chip do Sinclair, resultando num "parece que é, mas não é, vulgo Denorex" — acaba causando tamanha confusão para os usuários que desencoraja qualquer tentativa de desenvolvimento real de um software de qualidade.

Não fosse isto suficiente, basta o acontecimento de uma feira ou uma exposição de microcomputadores, e logo uma dezena de novos modelos são lançados na praça, alguns chegando ao cúmulo de serem lançados mais de uma vez. São os chamados "microfeira", cujas trajetórias, do lançamento ao aparecimento efetivo nas lojas, são muitas vezes longas a ponto deles se perderem no caminho. O consumidor em potencial, que já tinha dúvidas quanto ao equipamento a escolher, fica ainda mais confuso.

Estamos passando por uma fase de mistificação do hardware, que sem dúvida nos levará a um beco sem saída. Exemplos como Visicalc, Wordstar, etc., demonstram que o lampejo que cria o software independe do equipamento, pois a utilização é muito mais importante que o meio. De fato, hoje existem versões para todos os tipos de equipamentos dos melhores softwares criados internacionalmente nos últimos anos. O que precisamos são mecanismos e incentivos que preservem a atividade CRIATIVA, não importando que ela incida sobre um IBM PC, um Apple, TRS, ou Sinclair.

Alda Surenus Campos

**Atenção leitores:
MICRO SISTEMAS
mudou!**

Estamos atendendo agora nos seguintes endereços:

RIO DE JANEIRO — Av. Presidente Wilson, 165/grupo 1210, Centro, RJ, CEP 20030 — tels.: (021) 262-5259, 262-6437 e 262-6306.

SÃO PAULO — Rua Oliveira Dias, 153, Jardim Paulista, SP, CEP 01433 — tels.: (011) 853-7758, 881-5668 e 853-3800.

Micro Sistemas

Editor/Diretor Responsável:
Alda Surenus Campos

Diretor Técnico:
Renato Degiovani

Assessoria Técnica: Roberto Quitto de Sant'Anna; Luiz Antonio Pereira; Orson Voerckel Galvão; Carlos Alberto Diz.

Redação:

Edna Araripê (subeditoria); Cláudia Salles Ramalho; Denise Praganá; Graça Santos; Maria da Glória Esperança; Ricardo Inojosa; Stela Lachtermacher.

Colaboradores: Akeo Tanabe; Amury Moraes Jr.; Antonio Costa Pereira; Carlos Alberto Diz; Evandro Mascarenhas de Oliveira; Iva D'Aquino Neto; João Antonio Zuffo; João Henrique Volpini Mattos; Jorge de Rezende Dantas; Luciano Nilo de Andrade; Luis Lobato Lobo; Luiz Carlos Eiras; Marcus Brunetta; Paula Salles Mourão; Robson Vilela; Rudolf Horner Jr.

Revisão: Maria Christina Coelho Marques

Diagramação: Leonardo A. Santos

Arte Final: Vicente de Castro; Pedro Paulo S. Coelho

ADMINISTRAÇÃO: Janete Sarno

PUBLICIDADE

São Paulo:
Natal Calina
Contatos: Eloisa Brunelli; Marisa Ines Coan; Paulo Gomide.

Rio de Janeiro:
Elizabeth Lopes dos Santos
Contato: Regina de Fátima Gimenez

Minas Gerais:
Representante: Sidney Damings da Silva
Rua dos Caetés, 530 — sala 422
Tel.: (031) 201-1284, Belo Horizonte.

CIRCULAÇÃO E ASSINATURAS:
Ademar Belan Zochio (RJ)
Janio Pereira (SP)

DISTRIBUIÇÃO:
Fernando Chinaglia Distribuidora Ltda.
Tel.: (021) 268-9112

Composição:
Gazeta Mercantil S/A Gráfica e Comunicações

Fotolito:
Organização Beni Ltda.

Impressão:
Editora Vecchi S. A.

Assinaturas:

No país: 1 ano — Cr\$ 20.000,00

Os artigos assinados são de responsabilidade única e exclusiva dos autores. Todos os direitos de reprodução do conteúdo da revista estão reservados e qualquer reprodução, com finalidade comercial ou não, só poderá ser feita mediante autorização prévia. Transcrições parciais de trechos para comentários ou referências podem ser feitas, desde que sejam mencionados os dados bibliográficos do MICRO SISTEMAS. A revista não aceita material publicitário que possa ser confundido com matéria redacional.



MICRO SISTEMAS é uma publicação mensal do



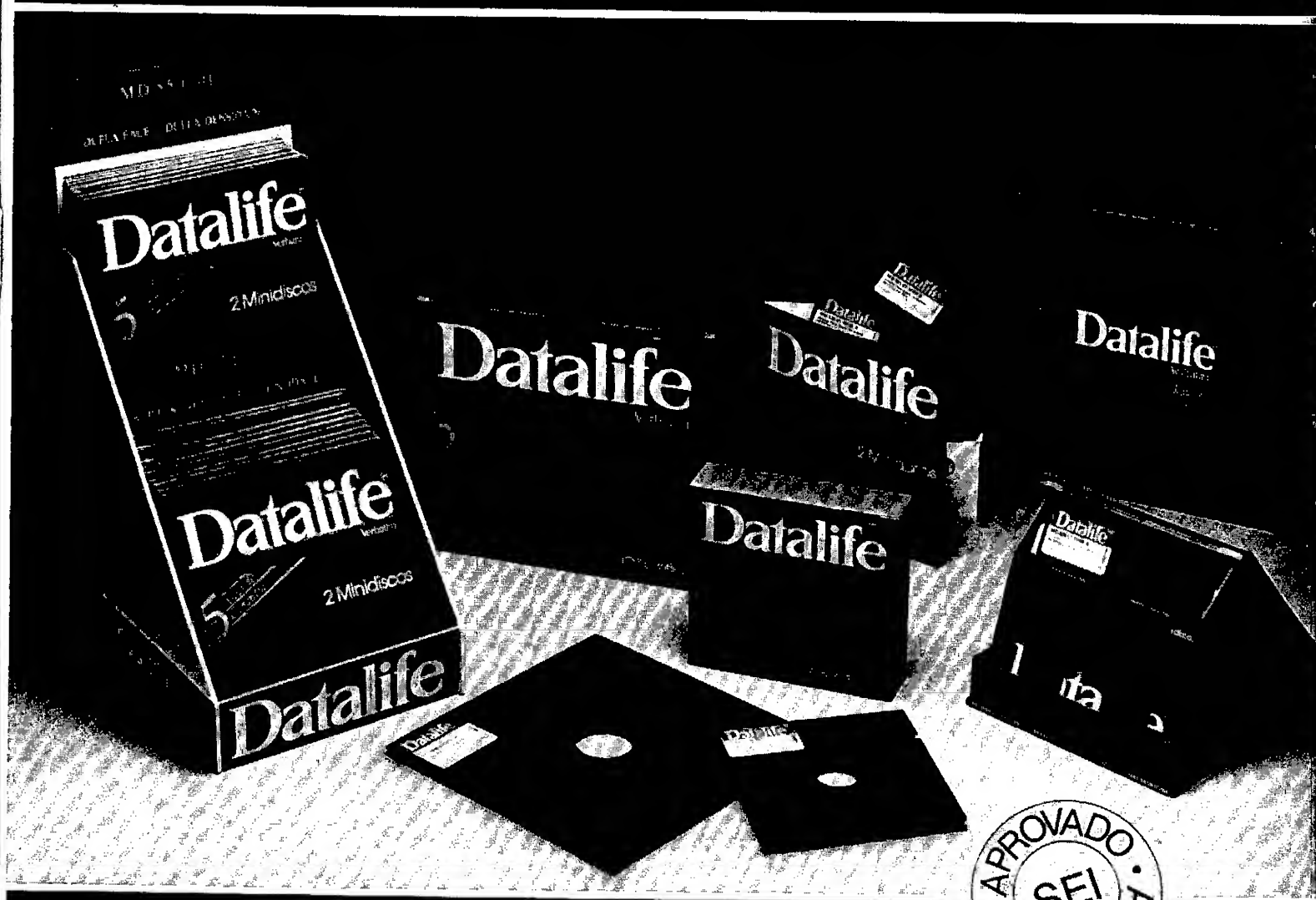
Análise, Teleprocessamento
e Informática Editora Ltda.

Endereços:

Rua Oliveira Dias, 153 — Jardim Paulista — São Paulo/SP — CEP 01433 — Tels.: (011) 853-3800, 853-7758 e 881-5668.

Av. Presidente Wilson, 165 — grupo 1210 — Centro — Rio de Janeiro/RJ — CEP 20030 — Tels.: (021) 262-5259, 262-6437 e 262-6306.

Datalife tem resposta para tudo.



E a Memphis responde junto.

Os Disquetes Datalife sempre têm um modelo exato para cada tipo de necessidade, seja ela para controles e decisões empresariais, assuntos pessoais e profissionais ou lazer com microcomputador.

Memphis®

São os mais vendidos no mundo e produzidos pela empresa que mais investe em pesquisa e desenvolvimento de produtos, líder mundial na tecnologia de mídia magnética flexível.

Tudo isso está a seu dispor na Memphis.

A Memphis responde, com a segurança de que você pode comprar

seus Disquetes Datalife num local onde eles têm pai, mãe, residência conhecida, e garantia de 5 anos. A Memphis tem estoques para entrega imediata.

Verbatim®

REVENDEDORES E FABRICANTES INTERESSADOS: (011) 262-5332

CENTRAL DE VENDAS: Av. Arnolfo Azevedo, 108 - São Paulo - Tel. (011) 262-5577 - Telex. (011) 34545 - FILIAL RIO DE JANEIRO: Praia do Flamengo, 66 Bloco B-Cj. 1519 - Tels. (021) 225-3469 e (021) 205-3849 - MATRIZ: Av. Angélica, 35 - São Paulo - Tel. (011) 826-9500 - REPRESENTANTES • BA - Salvador (071) 241-6369 • DF - Brasília (061) 223-3330 • MG - Belo Horizonte (031) 442-9472 • PR - Curitiba (041) 222-4831 • RS - Porto Alegre (0512) 25-9273 • SC - Florianópolis (0482) 132 ramal 15 • SP - Bauru (0142) 228-4305 • SP - Jundiaí (011) 434.3199 • SP - Presidente Prudente (0182) 22-7999

cartas

O sorteado deste mês, que receberá uma assinatura de um ano de MICRO SISTEMAS, é José Minerva de São Paulo.

ARQUIVAR DESENHOS

Tenho uma dúvida com relação ao programa "Desenhe no vídeo", publicado em MS nº 27: tentei, como o programa sugeria, arquivar o desenho usando a rotina "Arquive no vídeo", publicada em MS nº 23, na Seção Dicas. Mas dessa forma só seria possível arquivar um único desenho ou tela. Então, tentei passar o desenho depois de arquivado para uma variável, aproveitando a Dica "Inicializando variáveis", de MS nº 26, mas não sei porque isto não foi possível.

Gostaria que vocês publicassem uma rotina capaz de arquivar a imagem em uma variável qualquer, de forma que fosse possível arquivar vários desenhos em variáveis distintas. Se possível, gostaria que me informassem qual o endereço onde é arquivada a imagem naquela rotina "Arquive no vídeo".
Marcelo Batista Silveira
Osasco - SP

Nós também, Marcelo, gostaríamos de ter esta rotina que arquivasse vários desenhos em variáveis. Seria ótimo, mas infelizmente não temos. Talvez algum leitor já a tenha desenvolvido e nos mande, quem sabe? Vamos torcer que sim. Com relação ao endereço onde é arquivada a imagem na rotina "Arquive no vídeo", anote aí: é a partir de 31000.

EXPANSÃO NO CP-200

Recentemente comprei um CP-200 e já estou com vontade de aumentar sua capacidade de memória para mais de 16 K, se possível. Como sei que poderei contar com a ajuda de vocês, gostaria que me indicassem o melhor meio para fazer esta expansão, pois no manual que acompanha o micro não existem muitas informações.
Luís Paulo Neri de Souza
Mutuípe-BA

Como é de praxe, Luís Paulo, mandamos sua carta para a Prologica, e esta respondeu o seguinte:

"Com relação à consulta do leitor Luís Paulo, gostaríamos de esclarecer-lhe que a memória do CP 200 é de 16 K de RAM. Até o presente momento, não estamos cogitando em oferecer uma expansão de memória para o CP 200, o que não significa que isto não venha a acontecer no momento correto.

Se o leitor conhecer o microprocessador Z80A, terá a possibilidade de aumentar a capacidade de memória do micro, uma vez que, através da saída lateral do CP 200, tem-se acesso aos sinais deste microprocessador".
Eng. Dan Guinsburg
Gerente de Produtos-Computadores Pessoais da Prologica

IBAM

Solicitamos o endereço do Instituto Brasileiro de Administração Municipal - IBAM,

pois temos interesse numa informação veiculada em MICRO SISTEMAS nº 25, Seção Bits (Strings).

Jocymar Geraldo Lyra
Gerente Comercial da Digicap

O IBAM fica no Largo do IBAM, nº 1, Humaitá, Rio de Janeiro - RJ, CEP 22282. E o telefone do Setor de Promoção é (021) 266-6622 R. 292.

CURSOS

Escrevo-lhes para, antes de tudo, parabenizá-los pelo excelente trabalho que vem sendo feito por vocês. Gostaria também de apoiar a idéia do leitor Jorge Francisco Salazar, de Porto Alegre, que foi publicada na Seção Cartas de MS nº 27: criar uma seção com tabelas de preços dos micros e de videogames.

Gostaria ainda que vocês publicassem onde se pode fazer bons cursos de computação nas principais capitais brasileiras.
Denise Hemerly
Rio de Janeiro - RJ

Antes de mais nada, Denise, a equipe agradece os elogios: muito obrigada e bem-vinda à MS. As suas sugestões foram anotadas, mas com referência aos Cursos, já publicamos uma extensa matéria sobre este assunto em MS nº 18, página 60. Dê uma espiada. Você também pode ler nos números de MICRO SISTEMAS a Seção Cursos, bem como observar os diversos anúncios sobre cursos veiculados em nossa revista, ok?

DICAS PARA O DGT-100

Tenho um DGT-100, mas estou com dois problemas:

1º - como desativar o botão RESET, pois a tecla BREAK não tem mistérios;
2º - como, através de um programa em linguagem de máquina, conseguir chamar uma rotina, ou sub-rotina, em BASIC.

No primeiro caso, fiquei animado quando li, na página 18 de MS nº 26, o artigo "Programas trancados e selados", mas decepionei-me, pois o artigo era para micros compatíveis com o Apple. Deve haver algum meio de desativar o RESET no Digitus; tudo indica que seja utilizada a região de comunicação, mas não consigo descobrir como.

O meu programa DGEDAS (Editor Assembler da Digitus) desativa o RESET. Telefonei para a Digitus e o Sr. Alexandre disse-me que a tecla não é desativada. Não acredito, pois o programa DGEDAS, da própria Digitus, desativa-o. Será que vocês podem me dar uma dica?

No segundo caso, quero fazer a chamada, por um programa em LM, de uma rotina em BASIC. O livro "Microsoft BASIC decoded and other mysteries", na página 30, quando trata de BASIC Functions, apresenta a sub-rotina JP 1EB1-GOSUB como "capaz de fazer com que uma sub-rotina em BASIC seja chamada por uma sub-rotina em Assembler", e dá, inclusive, um exemplo com a função TRON do BASIC. Apesar dos meus esforços, não consegui que funcionasse. Já segui todos os passos dados no programa-exemplo e não consigo rodá-lo, pois sempre apresenta erro. Como posso fazer isso?
José Eduardo de Oliveira e Cruz
Rio de Janeiro-RJ

Infelizmente, José Eduardo, não podemos ajudá-lo. Mas talvez algum leitor mais

curioso e proprietário de um DGT-100 tenha como auxiliá-lo. Vamos fazer coro com o seu pedido, e esperar que algum leitor possa colaborar com o seu problema.

MS AGRADECE

O mundo dos micros é simplesmente fabuloso, o potencial inerente a esses "bichinhos" virou totalmente a minha cabeça.

(...) De todas as revistas especializadas que pesquisei, essa foi a única a chamar minha atenção; portanto, quero parabenizar os responsáveis e incentivá-los a manter o nível, que se tem mostrado excelente.

Através da Seção BITS, tomei conhecimento de um curso de BASIC que pode ser ministrado, ou seja, desenvolvido pelo interessado em sua própria residência. Nem é preciso dizer que já mandei um comunicado à SDI - System Design Ltda., solicitando as devidas informações. Desde já agradeço aos integrantes de MICRO SISTEMAS o apreço que têm por aqueles que estão tomando conhecimento dessa maravilhosa área de atuação.

Mostrei a revista a alguns amigos que também gostam da matéria e fiquei muito feliz com o pronunciamento de cada um deles, uma vez que se mostraram favoráveis ao conteúdo de MICRO SISTEMAS. Os meus parabéns a todos vocês, e continuem se esforçando para que a circulação desse periódico se mantenha na ativa. Respeito plenamente a opinião do nosso caro amigo leitor Mário Gomes da Silva, que citou as seguintes palavras na Seção Cartas: "Uma revista de alta competência deve ser sempre elogiada e falada. Como todos dizem, é uma revista MADE IN BRAZIL".
Valmir Rodrigues da Silva
São Bernardo do Campo-SP

Gostaria de parabenizá-los pela MICRO SISTEMAS nº 28, de janeiro de 84, que realmente trouxe programas muito interessantes para todos os tipos de microcomputadores. Gostaria também de parabenizar o autor do programa "Pescaria em águas eletrônicas", o Sr. Nelson Hisashi Tamura, que, usando a criatividade e sua experiência, fez um ótimo programa. Achei também muito bom o programa "Aventuras em Serra Pelada", só que não me animei a digitá-lo. Parabéns e que a revista MICRO SISTEMAS se desenvolva por muitos e muitos anos.
Henrique A. Vianna
Pelotas - RS

Sou há alguns meses proprietário de um TK-85, o qual comprei para obter auxílio no cálculo estrutural em Engenharia Civil. MS conseguiu despir por completo a impressão séria e antipática que eu tinha dos microcomputadores pessoais, conduzindo-me ao fascinante mundo dos jogos... Envio-lhes meus sinceros parabéns e que continuem a publicar aplicativos e jogos (principalmente em Assembler) para a linha Sinclair.
Dilson Lara Júnior
São José dos Campos - SP

Parabenizamos esta conceituada revista pela decisão de incluir uma Seção dedicada às aplicações do micro no radioamadorismo. Sentimo-nos duplamente contentes: pela idéia e pela escolha do responsável, Roberto Quito de Sant'Anna, PY 1 DWM, nosso companheiro, Awards Manager e Membro do Conselho Deliberativo do CWRJ (...).
Ronaldo Curi Gismondí
Secretário Geral do CWRJ

Como usuário de micros da linha Sinclair há dois anos, parabeno MICRO SISTEMAS por dedicar tantas páginas a este sistema, que não só tem o maior número de aficionados entre todos os outros, como é o mais acessível, realístico e possui um universo muito maior (tanto em software quanto em hardware).
Adriano Giusfredi de A. Botelho
Amparo - SP

SUGESTÕES

Sou leitor de MICRO SISTEMAS e acho que é a revista mais completa sobre computadores do Brasil. Gostaria de sugerir que vocês fizessem um curso de BASIC e um de Assembler especialmente para crianças.

Gostaria também que vocês fizessem todo mês a avaliação de um micro, e ainda que a cada seis meses publicassem uma tabela com os principais micros e seus preços.

Jorge Pablo Z. Rivera
Salvador - BA

Quero sugerir a publicação de um Monitor Assembler (do tipo que foi publicado em MS nº 23) para as linhas TRS-80 e Apple. Gostaria também que voltassem a publicar artigos e programas para a TI-59.

Sugiro também que publiquem uma reportagem sobre o TK-2000 Color, com todas as suas características: o microprocessador usado, o sistema operacional, quais periféricos que aceita e onde podem ser encontrados, se existem programas em cartuchos ROM... Tudo, enfim, sobre este equipamento.
Miguel Angelo Henzo
São Paulo - SP

Visando ao aperfeiçoamento e engrandecimento da revista, gostaria de fazer a seguinte sugestão: várias revistas de automóveis possuem uma seção sobre preços de veículos. Que tal se MS tivesse também uma seção parecida, onde encontraríamos preços e outras características dos micros nacionais? A seção poderia chamar-se "Mercado dos Micros".
Marco Túlio de O. Valente
Viçosa - MG

Gostaria que a revista colocasse uma Seção Dicas só para equipamentos da linha Sinclair, pois nós, os usuários destes equipamentos, sentimos muito a falta de publicações realmente úteis, principalmente no sentido de aprender a programar em linguagem de máquina. E vocês poderiam, na medida do possível, colocar também na revista um curso em linguagem de máquina para equipamentos da linha Sinclair, assim como foi feito com o curso de Assembler.
Edison Bueno da Silva
Jundiaí - SP

Não abusando da boa vontade com a qual Vs. Sas. têm atendido aos seus leitores, gostaria, se possível, de ver publicados programas em BASIC na área contábil, como: lançamentos contábeis, contas a receber e a pagar, depreciações, balancete, balanços etc.
Divo Machado dos Santos
Sorocaba - SP

Se possível, coloquem juntamente com o nome dos autores dos programas os respectivos endereços e telefones para um maior e mais rápido esclarecimento de supostas dúvidas existentes nos programas que eles editaram.
Eduardo Lazzarin Marques
São Paulo - SP

Sugiro a publicação de programas aplicativos compatíveis com o TK-85.
Murillo E. Ariz
Porto Alegre - RS

Fui a uma banca de jornais procurar alguma coisa sobre microcomputadores. O jornalista me ofereceu cerca de sete revistas, vistoriei todas e selecionei a melhor: era a MICRO SISTEMAS, os artigos publicados eram exatamente o que eu procurava. Parabeno todos os produtores desta revista, uma publicação sensacional. Desejava apenas que vocês publicassem mais programas para computadores da Linha Sinclair.
Marcelo Thomaz
Santo André - SP

Gostaria de sugerir que publicassem o Micro Mercado (tanto para micros quanto para impressoras e outros periféricos) por bimestre ou trimestre. Acho também que tabelas de funções de várias linguagens seriam de grande utilidade para os principiantes. E, finalizando, sugiro também a publicação (bimestral ou trimestralmente) de todos os programas lançados no mercado.
Nelson Junzi Nagamine
Embu - SP

Acho que vocês poderiam criar uma Seção voltada para os micros compatíveis com o TRS-80, pois, proprietário que sou de um DGT-100, fiquei com "ciúmes" do Apple por causa da Seção Sidra (muito bem feita pelo Sr. Horner).
Robinson S. Pereira
Rio de Janeiro - RJ

Solicitaria aos caros amigos que, quando da publicação de um programa, fizessem a necessária explicação sobre como se efetua o seu funcionamento (os iniciantes, como eu, ficam sem opção).
José Minerva
São Bernardo do Campo - SP

Descobri que MICRO SISTEMAS era a revista de que precisava para o meu micro. Com ela já rodei vários programas que me ensinaram muitas coisas. Gostaria, entretanto, de dar uma sugestão: publiquem um curso de linguagem de máquina para a linha Sinclair.
Silvio Ferraz
Piracicaba - SP

Gostaria que fosse fundado um Clube de Software pela MS para trocas de idéias, conhecimentos e programas. Acho também que um curso de BASIC ministrado por vocês seria interessante.
Wander M. Martins
Itajubá - MG

Possuo um micro VIC-20, e para encontrar artigos sobre ele é preciso comprar revistas importadas. Por isso, faço uma sugestão: que em sua revista seja publicada uma página destinada a micros importados como o meu. Acredito que a vendagem aumentaria muito, pois há um grande número de pessoas que possuem micros importados. Entendo que sua revista procure fortalecer a indústria nacional, porém acho que o dinheiro que se gasta comprando revistas importadas poderia reverter para o mercado nacional.
Marcos André O. Paraíso
Rio de Janeiro - RJ

Envie suas sugestões para MICRO SISTEMAS. Elas serão anotadas em nossa pauta e procuraremos, na medida do possível, viabilizá-las.

COLABLOC

LANÇAMENTO NACIONAL

A COLADORA DE BLOCOS DE PAPEL DA LAURENTI.



COLA EM MENOS DE 30 SEGUNDOS.

Colabloc - 1ª coladora nacional com tempo total de operação máximo de 30 segundos.

Produto de mesa, compacto, simples e seguro, ideal para escritórios que não exijam sistemas complexos de encadernação.

Perfeito para atender às necessidades de usuários de computadores e gráficas, cujo volume de trabalho de encadernação não seja grande.

Executa um trabalho limpo por não necessitar de contato manual com a cola. Não exige instalações especiais, sendo apenas necessário uma tomada monofásica de 110 volts 50/60 Hertz.

 **laurenti**

EQUIPAMENTOS PARA PROCESSAMENTO DE DADOS LTDA.

Matriz: Rua Theodureto Souto, 308 - Cambuci - CEP: 01539
PABX: 270-8244 - Telex: (011) 36305 - São Paulo - SP
REPRESENTANTES TÉCNICOS E COMERCIAIS EM TODO O BRASIL.

A rotina aponta-erros

Carlos Alberto Diz

O maior trabalho na confecção de um programa não está em escrevê-lo, mas sim em conseguir rodá-lo em sua totalidade sem erros! Qualquer pessoa que tenha se proposto a escrever algo mais complexo do que um simples *loop* para imprimir os dígitos de 0 a 9 certamente já descobriu esta grande verdade. Segundo a famosa *lei de Murphy*, sempre que puder haver um erro ele estará lá, e o corolário acrescenta que sempre haverá um erro a mais do que o esperado.

A técnica de depurar um programa de seus inevitáveis erros chama-se *debugging*, e adquirir habilidade nesta tarefa deve ser realmente o principal objetivo de qualquer um que deseje alcançar níveis de complexidade mais elevados em seus programas. *Debugar* programas, como se diz em *computês*, é por vezes uma árdua aventura, pois geralmente o erro se esconde no meio das linhas entre vírgulas e aspas, escapando assim até ao olho mais atento durante pelo menos as primeiras 132 revisões da listagem. Existem, adicionalmente, dois fatores agravantes: 1) nem todos possuem uma impressora (no Brasil um investimento de porte considerável) para listar o programa no papel e facilitar um pouco a humilhante busca do malfadado erro. Não possuindo tal instrumento, o infeliz programador não tem nenhuma alternativa senão passar horas olhando para o vídeo até encontrá-lo, ficar estrábico ou desistir;

2) muitos programadores (os mais avançados por certo) descobrem que é possível obter-se alguma economia de memória juntando vários comandos numa só linha, separando-os com ; como nos ensina qualquer manual de BASIC. Se é verdade que isto economiza memória, também devemos reconhecer que esta técnica torna muito mais vitorioso o esforço do erro em fugir ao olho atento (no início, pelo menos) do programador.

No meio disso tudo, é preciso lembrar, no entanto, que o interpretador sempre nos ajuda um pouco com suas rotinas de *error-trap* e mensagens que nos dizem quando ocorre um erro, em qual linha e de que tipo se trata. Mesmo assim, saber em que linha há um erro pode limitar o campo de busca, mas a

mensagem sobre o tipo de erro é freqüentemente ambígua ou não faz sentido algum mesmo. Além disso, no caso de linhas compactadas, cheias de instruções, o interpretador não diz exatamente onde está o erro (e às vezes existem mais de um), o que nos leva a concluir que o interpretador *torce* pelo erro!

DA PROCURA AO GRANDE ACHADO

Lembro-me de ter encontrado, em algum momento de minha vida, um interpretador de BASIC (se não me angano num HP-2000F) que, quando detectava um erro, era tão gentil que colocava uma flechinha embaixo do comando errado ao listar a linha *culpada*. Lembrando-me disto, e remechendo no meu Apple há algum tempo atrás, descobri alguns endereços interessantes que, ocorrendo um erro, podem ajudar a encontrá-lo.

Os endereços (em decimal) 218 e 219 contêm, respectivamente, o byte menos significativo e o byte mais significativo correspondentes ao número da linha onde existe um erro. É daqui que o interpretador tira a informação que tão graciosamente nos proporciona quando diz: "erro na linha tal". E ainda, e isto o interpretador não nos revela, os endereços 220 e 221 (também decimais) contêm o endereço do *statement* na linha que causou o erro (se este ocorreu no primeiro *statement*, será necessário acrescentar 5 a este valor), ou seja, diz em que parte da linha procurar. Apesar de não ser a coisa mais clara do mundo, esta informação pode ser usada de várias formas para localizar o erro.

Descobertos estes endereços, e depois de muito trabalho ingrato, consegui fazer uma rotina que mais ou menos me informava a posição de um erro na linha, mas a rotina não era realocável e continha valores que variavam em função do tamanho do programa ao qual estivesse *pendurada*. Isto fazia com que cada vez que o programa tivesse de ser alterado fosse necessário alterar também estes valores na rotina. Enfim, era tão complicada e trabalhosa de usar que, em minha opinião, os benefícios que trazia não compensavam o trabalho que dava.

10 ONERR GOTO 60000:EPT=63:T=256

```
60000 E=PEEK(220)+T*PEEK(221):E=E+5*(PEEK(E)=0):V=PEEK(E):POKEE,EPT:L$=RIGHT$(  
"0000"+STR$(PEEK(218)+T*PEEK(219)),5):L=PEEK(121)+T*PEEK(122)+49:FOR I=1  
TO 5:POKE L+I,ASC(MID$(L$,I,1)):NEXT LIST 00000:POKE E,V:POKE 216,0:RESUME
```

Figura 1

Recentemente, porém, chegou às minhas mãos uma rotina em Applesoft — cujo autor desconheço — que é um verdadeiro *ovo de colombo*, pois traz uma solução muito engenhosa para a utilização da informação que existe naqueles endereços que mencionei. É também um ótimo exemplo de código *auto-modificante*, uma técnica capaz de *arrepiar* qualquer fanático de programação estruturada, pois é realmente a antítese da estruturação.

Rapidamente: código *auto-modificante* é um código que, ao rodar, altera um ou mais comandos nele contidos, ou seja, o conteúdo de uma linha de programa é alterado uma ou mais vezes ao rodarmos o próprio programa. Algo muito parecido com levantar-se pelos laços dos sapatos. Tal recurso, impossível de ser concebido na mente organizada e rígida dos fanáticos da estrutura, permite fazer *coisas do arco da velha*, mas como tudo que é potente é também perigoso, pode, quando mal utilizado, destruir programas e levar o programador desprezado à loucura.

Outra grande vantagem da rotina é o fato dela ser realocável (ao contrário da que eu havia *bolado*), isto é, ela pode ser *pendurada* em qualquer lugar e em qualquer programa sem que sua posição ou alterações no programa afetem seu funcionamento (da rotina). Em outras palavras: a rotina não contém endere-

ços implícitos ou explícitos em valores absolutos e sim somente em valores relativos.

COMO USAR A ROTINA

Tudo que vocês precisam fazer é digitar o código em seus micros (somente Apple compatíveis, por enquanto) na exata forma como está listado na figura 1. Mas atenção: é importante que a forma seja exatamente idêntica, especialmente no que concerne à linha 60000, pois de outra maneira certos valores precalculados de *displacement* poderão perder sua validade.

Inserindo o programa a ser *debugado* entre as linhas 10 e 60000 e rodando-o normalmente, caso ocorra um erro você receberá, além da costumeira mensagem de erro, uma listagem da linha condenada na qual terá sido incluído um ? no começo do *statement* que contém o erro.

Se você não gostar dos números 10 e 60000 que eu atribuí às linhas, mude-os à vontade, contanto que o seu programa fique entre as duas. Nunca modifique, porém, o conteúdo da linha 60000, a menos que você saiba o que está fazendo, senão corre o risco de ver o seu programa alterado de maneira imprevisível e incrível! Se também não lhe agradar (mas vai ser difícil assim...) o ?, substitua-o por um ! ou um > ou o



Convite

A Compumicro Informática Empresarial tem o prazer de convidar V. Sa. para uma apresentação do microcomputador NEXUS 1600, com sua avançada tecnologia e inúmeras aplicações para imediata utilização.

Inteiramente compatível com os equipamentos IBM PC/XT, o NEXUS 1600 é oferecido pela Compumicro dentro de uma filosofia de implementação de soluções, envolvendo a seleção de Hard/Soft, treinamento personalizado e implantação, podendo abranger também sua ligação com equipamentos de grande porte.

Confirme sua visita e venha conhecer o mais moderno microcomputador disponível nos dias de hoje.

compumicro
INFORMÁTICA EMPRESARIAL LTDA.

Rua Sete de Setembro 99/11.º andar RJ.

Tel.: PBX (021) 224-7007

Analizando a rotina

Linha 10

- Estabelece que, em caso de erro, o controle deverá passar para a linha 60000.
- Atribui à variável EPT o valor 63, que corresponde a ASC(?).
- Atribui à variável T o valor 256 (número *compu-cabalistico*).

Linha 60000

- Atribui à variável E o valor correspondente ao *pointer* de execução, ou seja, a posição de começo do *statement* que contém o erro.
- Caso o erro esteja no primeiro *statement* da linha, fazendo com que haja um CHR\$(0) na posição E, ou seja, PEEK(E)=0, então acrescenta 5 ao valor de E tomando-o correspondente à posição do primeiro caráter do primeiro *statement* da linha.
- Atribui à variável V o valor contido na posição E, isto é, copia para A o ASC do caráter apontado por E (isto para repô-lo em seu lugar no fim de nossa rotina).
- Finalmente, coloca na posição apontada por E (no lugar do caráter que mencionamos acima) o valor EPT, que no nosso caso corresponde ao ASC de ?, certo?
- Atribui a L\$ os primeiros cinco caracteres contados a partir da direita do que resulta da soma de 0000 com a transposição para a *string* do valor contido nos endereços 218 e 219, ou seja, o número da linha, garantindo sempre ter 5 dígitos e colocando zeros à esquerda quando este for inferior a 10000.
- Atenção, pois aqui está o grande golpe de gênio e o detalhe de auto-modificação que mencionei: atribui à variável L o valor 49 +

o valor contido nos endereços 121 e 122. Estes endereços apontam em qualquer momento para a posição inicial do *statement* que está sendo executado no momento. Neste caso, quando for executada a instrução, eles apontarão para o : que precede L=PEEK(121)+T*PEEK(122). Somando 49 ao valor deste *pointer*, obtemos então a posição do primeiro 0 no *statement* LIST 00000 um pouco mais adiante na mesma linha. Se contarmos quantos caracteres há entre o primeiro : nesta linha e o primeiro 0 em LIST 00000, lembrando que os comandos de BASIC correspondem a um só caráter, veremos que há exatamente 49.

- O *loop*, executado 5 vezes, faz nada mais nada menos que substituir os 00000 no *statement* LIST 00000 (seguinte) pelo conteúdo de L\$, ou seja, o número da linha mais alguns zeros à esquerda conforme necessário. Em outras palavras: a instrução LIST 00000 originária do programa é alterada para LIST XXXXX, onde XXXXX é o número da linha do erro. Depois de executado o *loop*, se você listar o programa verá que realmente foi alterado o código da linha 60000!

- O *statement* LIST XXXXX, quando executado, faz exatamente o que diz a linha número XXXXX que contém o erro, mas como esta linha havia sido também modificada (colocamos um ? no começo do *statement criminal*), ela será listada junto com a mensagem de erro, contendo um ? para nos indicar onde encontra-se o erro.

- Finalmente, coloca-se de volta no lugar do ? o que lá havia anteriormente e que, lembrem-se, havíamos guardado em V.
- Reseta-se o *pointer* em 216, usado pela rotina de *error trap*.

que você quiser, alterando o valor atribuído a EPT na linha 10. Por exemplo:

```
10 ONERR GOTO 60000:EPT=33:T=256
```

fará com que apareça um ! denunciando o *statement* culpado pelo erro.

Você perceberá ainda que, quando o erro estiver do segundo *statement* da linha para frente, o ? aparecerá no lugar do : que separa o *statement* culpado do *statement* precedente; porém, quando o erro estiver no primeiro *statement* da linha, o ? aparecerá no lugar do primeiro caráter da linha, logo após o número da mesma.

Este pequeno defeito deve-se a razões intrínsecas à forma como o interpretador guarda as instruções do programa e em particular ao fato de que toda linha começa com um número para o qual são reservados 5 bytes, vindo logo a seguir uma instrução, seja um nome de variável, um *token* ou sei lá. Quando o interpretador está executando o primeiro *statement* de uma linha, o *pointer* de sequência de execução aponta para o primeiro byte do número da linha (5 à frente da posição do primeiro caráter da linha depois do número), enquanto que na execução do segundo *statement* de uma linha em diante, este *pointer* aponta para a posição onde está contido o : que separa este *statement* do anterior. Como a rotina coloca o ? na posição apontada por este *pointer* (ou esta +5 quando se trata do primeiro *statement*), explica-se a razão do ? estar onde está.

Vamos entender melhor com um exemplo? Inclua esta linha:

```
10000 PRINT "O ERRO ESTA' NO PROXIMO
STATEMENT"? PRNT "A PALAVRA PRINT
ESTA' MAL ESCRITA"
```

e rode o programa. Se você tiver feito tudo direitinho, na tela aparecerá:

```
SYNTAX ERROR IN LINE 10000
```

```
10000 PRINT"O ERRO ESTA' NO PROXIMO
STATEMENT"? PRNT "A PALAVRA PRINT
ESTA' MAL ESCRITA"
```

Note o ? no lugar do : antes de PRNT, indicando onde está o erro. Se isto não acontecer, revise a linha 60000 buscando erros de grafia ou diferenças com respeito à figura 1 e volte a tentar. Quando conseguir, liste o programa para ver como na linha 60000 agora tem LIST 10000 no lugar de LIST 00000.

Seguindo as instruções, eu garanto que funciona, e tenho certeza que será de grande ajuda naquelas longas seções de *debugging*.



Carlos Alberto Diz é formado em Engenharia Eletrônica pela Universidade de Dundee, Escócia, e possui Mestrado em Administração de Empresas pelo INSEAD - Instituto Europeu de Administração de Empresas, Fontainebleau, França. Atualmente é sócio-garante da Compusystems do Rio de Janeiro, uma system-house dedicada à consultoria informática e confecção de software sob medida.



PROCURE QUEM
REALMENTE ENTENDE.

MICROMAQ

R. Sete de Setembro, 92 - Lj. 106
Tel.: 222-6088 - Rio de Janeiro

**POR QUE NÃO TUDO EM
UM SÓ LUGAR?**

Microcomputadores, Software, Publicações
Especializadas, Cursos e Manutenção de Equipamentos.

MICROCRAFT
MICROCOMPUTADORES LTDA.



- RAMCARD • SOFTCARD • VIDEOTERM • SOFTVIDEO SW • PROGRAMMER • PROTOCOL CARD • INTF. DISKS
- INFT. PRINT. • SATURN 128K RAM. • SATURN 64K RAM. • SATURN 32K RAM. • RANA QUARTETO • MICROMODEM II
- MICROBUFFER II • MICROCONVERTER II ■ MICRO VOZ II ■ ULTRATERM ■ ALF 8088 CARD
- A800 DISK CONT ■ MULTIFUNCTION CARD

MICROCRAFT MICROCOMPUTADORES LTDA.

ADMINISTRAÇÃO E VENDAS: AV. BRIG. FARIA LIMA, 1.664 - 3º ANDAR - CJ 316 - CEP 01452
FONES (011) 212-6286 E 815-6723 - SÃO PAULO - SP - BRASIL

Datalife

tem resposta para tudo



Inclusive para quem não precisa comprar 10 disquetes de uma vez.

Datalife apresenta seu álbum duplo: uma embalagem adequada para quem precisa de um ou dois disquetes agora e só vai precisar de outros daqui a algum tempo. Em vez de você ficar comprando disquete solto por aí, que passa de mão em mão, com o álbum duplo você tem toda a segurança que existe.

Ele contém apenas dois disquetes e vem lacrado.

E o que tem dentro, você já conhece:

Disquetes Datalife com certificação crítica, com desempenho muito além das normas tradicionais, e 100% livres de erros.

Disquetes Datalife - na embalagem normal ou no álbum duplo, a garantia total de um nome que é líder mundial em mídia magnética flexível.

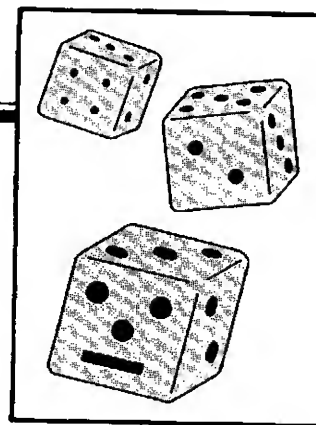


Verbatim®

Você acumula vários pontos... e de repente pode perdê-los todos! O jeito é confiar na sorte, não arriscar muito e deixar rolar os dados no PC-1500

Groan: os dados estão em jogo

José Roberto Cottim



Neste jogo de dados, que adaptei do *Level II BASIC* para o BASIC do PC-1500 da Sharp (Radio Shack PC 2), fiz questão de adicionar rotinas exclusivas do PC-1500 para deixar os orgulhosos possuidores deste computador de bolso mais orgulhosos ainda.

Você e o computador possuem dois dados para jogar. O programa começa perguntando ao jogador com quantos pontos ele gostaria que o jogo terminasse — qualquer valor entre 50 e 100 dará uma ótima partida, e ganha quem conseguir atingi-lo primeiro. Em seguida, o computador joga uma moeda para cima, sorteando quem dará início à partida.

Os dados podem ser passados de um para o outro de duas formas: espontaneamente ou de maneira mais drástica. Se em uma determinada jogada um de seus dados marcar 1, não só eles passarão para o adversário como também todos os pontos que você acumulou naquela rodada ficarão perdidos. E tem pior: se os dois dados marcarem 1, todo o seu placar irá a zero, além, é claro, de você perder a vez (você poderá perceber que o número 1 foi representado por uma *carinha triste*, pois é assim mesmo que você irá se sentir).

Para o computador escolher, na vez dele, se passará os dados para você ou se continuará a jogar, há uma rotina a partir da linha 900. Na

sua vez, logicamente, todo o seu bom senso e perspicácia se farão necessários, uma vez que o computador não é bobo e irá atrás de você se a distância entre os dois placares for muito grande. Os pontos de uma rodada, portanto, só passarão a fazer parte do placar geral quando você, ou o computador, passar os dados para o outro espontaneamente.

Ao final da partida o vencedor é premiado com uma mensagem, cujas letras geram uma *musiquinha cibernética* especial. Logo depois, o *display* é invertido por intermédio de uma rotina em linguagem de máquina.

É isso aí. Não perca tempo, digite logo a listagem e tenha boa sorte (você vai precisar dela!!!).

José Roberto França Cottim estuda Engenharia de Telecomunicações na Universidade Federal Fluminense. Trabalha na FOCO, firma promotora de feiras, exposições e congressos, e desenvolve sistemas para microcomputadores desde 1981.

Groan

```
100: "D" CLEAR : WAIT 0: RANDOM : CLS
110: A$ = "7F41656965417F": B$ = "7F41454151417F": C$ = "7F41454951417F"
115: D$ = "7D41554155417F": E$ = "7F41554955417F": F$ = "7F416B416B417F"
120: G$ = "0000000000": H$ = "0000000000": I$ = "001C1C1C00": J$ = "003E3E00"
125: K$ = "7F7F7F7F7F7F": M$ = "Os dados passam ": N$ = "para "
130: DATA 7,11,1,1,1,7,-1,0
135: RESTORE
140: INPUT "Final em quantos pontos ? ", W
150: P$ = "Voce": PRINT "Sorteando": Q$ = P$ + " comeca": Q = RND 2
160: IF Q = 2 LET Q$ = "Eu comeco"
170: WAIT 25: FOR I = 1 TO 5: PRINT ".": NEXT I
180: WAIT 200: PRINT Q$: Q$ = " jogando": T = 0: IF Q = 2 THEN 300
200: WAIT 50: CLS : F = 0: D = 1: PRINT P$: D$: A: T: GOSUB 400: GOSUB 700
210: IF F > 0 WAIT 100: PRINT M$: N$: "mi": GOTO 300
220: WAIT 0: CURSOR 0: PRINT "(J)oga ou (P)assa ?":
230: READ X: IF X = 0 RESTORE : READ X
240: READ Y, Z: FOR I = X TO Y STEP Z: GOSUB 250: NEXT I: GOTO 230
250: C = 118: R = 1: GOSUB 600
260: Q$ = INKEY$: IF Q$ = "" RETURN
270: IF Q$ = "J" THEN 200
280: IF Q$ = "<" THEN RETURN
290: CLS : A = A + T: WAIT 200: PRINT P$: " tem": A: GOSUB 600
295: F = 1: GOTO 210
300: T = 0: D = 2
310: WAIT 50: F = 0: PRINT "Eu": Q$: B: T: GOSUB 400: GOSUB 700
320: IF F > 0 THEN 350
330: GOSUB 900: WAIT 100: IF X PRINT "Jogarei novamente": GOTO 310
340: B = B + T: PRINT "Voce parar agora. Tenho": B: GOSUB 800
350: PRINT M$: N$: "voce": T = 0: GOTO 200
```

```
400: D = RND 6: S = RND 6
410: C = 135: R = 0: GOSUB 600: C = 145: R = 5: GOSUB 600: BPRINT 0
420: IF Q = 1 LET F = 1: GOSUB 500
430: IF S = 1 LET F = F + 1: GOSUB 500
440: IF F = 2 WAIT 0: PRINT "ZERO": FOR I = 0 TO 1: BEEP 1, 60 - I * 10, 400:
NEXT I: BEEP 1, 35, 600
450: RETURN
500: BEEP 1, 80, 100: RETURN
600: GOSUB C: BPRINT Q$(R): RETURN
700: T = T + 0.5: IF F > 0 LET T = 0
710: IF F = 2 LET Q(D) = 0
720: RETURN
800: IF Q(D) < 0 RETURN
810: R$ = "A FERA aqui ganh": S$ = "ou de novo":
V$ = P$ + " ganhou - Me": W$ = "ra sorte!!"
820: WAIT 0: IF B > 0 PRINT R$: S$: J = 29040
830: IF A > 0 PRINT V$: W$: J = 29104
840: FOR I = 0 TO 25: BEEP 1, PEEK (J + 1), 100: NEXT I
850: POKE J, 72, 118, 74, 0, 5, 189, 255, 65, 78, 78, 153, 8
851: POKE J + 12, 76, 119, 139, 6, 72, 119, 74, 0, 158, 18, 154
860: FOR I = 20 TO 1 STEP -1: CALL J: FOR K = 0 TO I: NEXT K: NEXT I: END
900: V = B + T: IF V > 0 THEN 960
910: IF W < 10 THEN 970
920: IF B > 0 LET L = T / 25: GOTO 950
930: IF V < 10 LET L = T / 35: GOTO 950
940: L = T / 30
950: IF RND 0 > L THEN 970
960: X = 0: RETURN
970: X = 1: RETURN
```

ALGARISMO: 0
CARACTER: 0

Programas que raciocinam

Nos artigos Inteligência Artificial I e II, publicados, respectivamente, nos números 29 e 30 de MICRO SISTEMAS, abordamos a importância das técnicas de programação de jogos para o desenvolvimento da Inteligência Artificial, exemplificando com um jogo inteligente para a linha TRS-80. Neste número falaremos sobre os sistemas especialistas, mais uma prova de que a Inteligência Artificial é capaz.

Antonio Costa • André Gurgel

Digamos que você queira escrever um programa capaz de efetuar uma investigação policial como aquelas que ocorrem em histórias de detetive. Seu programa pode realizar esta tarefa *raciocinando* como Sherlock Holmes ou como os detetives de Agatha Christie.

O raciocínio usado por Sherlock Holmes é denominado progressivo. Holmes parte de fatos e vai deduzindo outros fatos até chegar à solução do problema no qual ele está trabalhando. Já Hercule Poirot, o famoso detetive belga criado por Agatha Christie, usa o raciocínio regressivo. Poirot parte de uma lista de hipóteses e procura reunir fatos que lhe permitam concluir que uma das hipóteses é verdadeira.

Um dos maiores êxitos alcançados pela Inteligência Artificial foi a construção de programas que, usando raciocínio regressivo, conseguiram realizar trabalhos em medicina, geologia, química, engenharia de manutenção etc. Tais programas são conhecidos por sistemas especialistas.

O primeiro sistema especialista a conquistar o interesse do público foi o MYCIN, escrito em LISP pelo médico E. H. Shortliffe (o MYCIN é capaz de diagnosticar doenças infecciosas e indicar o tratamento). Nenhum sistema especialista, entretanto, causou mais entusiasmo do que o PROSPECTOR. A finalidade deste programa é localizar áreas apropria-

das para a prospecção geológica e seu bom desempenho nesta tarefa foi comprovado em 1980, quando ele descobriu importante mina de molibdênio no Estado de Washington.

As aplicações dos sistemas especialistas são inúmeras. A avaliação estática do Belle, por exemplo, é realizada por um sistema especialista. Para quem não sabe, Belle é um programa capaz de jogar xadrez quase a nível de mestre nacional. Roger Schanck está comercializando sistemas especialistas capazes de ajudar as pessoas a escrever testamentos e a preencher formulários de imposto de renda. A GE encomendou um programa para dirigir a manutenção de suas locomotivas. Um programa conhecido como ONCASYN é usado para fazer o acompanhamento de pacientes submetidos a quimioterapia. Em Portugal, L. Pereira e colaboradores construíram o ORBIS, um programa que entende Português e indica a melhor forma de usar recursos naturais. Poderíamos fornecer muitos outros exemplos, mas acreditamos que estes já são suficientes para convencer o leitor da importância dos sistemas especialistas.

A finalidade deste artigo é mostrar como os sistemas especialistas funcionam. Para isto, fornecemos um pequeno programa capaz de realizar uma investigação policial. Estudando este programa, o leitor entenderá o mecanismo do raciocínio regressivo.

Gostaríamos de frisar que os sistemas especialistas reais são centenas de vezes mais complexos do que o fornecido aqui. Nosso sistema, por exemplo, não consegue trabalhar com informações inexatas ou incompletas e isto o torna inútil para atacar problemas reais. Se o leitor necessita de um sistema especialista para uma aplicação real, aconselho-o a ler os vários artigos que saíram na revista *Artificial Intelligence* sobre o assunto.

TRANSFORME SEU COMPUTADOR NUM DETETIVE

O sistema especialista que iremos discutir trabalha com três listas: a lista de hipóteses, a lista de fatos e a lista de regras. No início da investigação, a lista de fatos pode estar vazia ou conter informações conhecidas de antemão. No decorrer da investigação, o programa perguntará se certas frases são verdadeiras ou falsas. Dependendo da resposta do usuário, ele colocará a frase ou a negação da frase na lista de fatos.

A lista de hipóteses contém, como seu nome indica, as hipóteses que o sistema deseja demonstrar ou refutar. A lista de regras, da qual há um exemplo na *Listagem 1*, é dividida em blocos denominados regras. Dentro de cada regra, a sentença que vem antes do condicional SE é chamada *consequente* e as sentenças que vêm depois do SE são chamadas *an-*

Listagem 1 - Regras e hipóteses

LISTA DE REGRAS

```
( ( (A CAMAREIRA ROUBOU O COLAR)
SE
(A CAMAREIRA PRECISAVA DE DINHEIRO)
(ERA FACIL VENDER O COLAR) )
( (A CAMAREIRA PRECISAVA DE DINHEIRO)
SE
(A CAMAREIRA GASTAVA MUITO)
(A CAMAREIRA GANHAVA POUCO) )
( (A GRALHA ROUBOU O COLAR)
SE
(A JANELA ESTAVA ABERTA)
(O COLAR ERA BRILHANTE)
(GRALHAS GOSTAM DE COISAS BRILHANTES) )
( (A JANELA ESTAVA ABERTA)
SE
(NAO ESTAVA CHOVENDO)
(ESTAVA FAZENDO CALOR) )
( (ESTAVA FAZENDO CALOR)
SE
(ERA VERAQ) )
( (OS CIGANOS ROUBARAM O COLAR)
SE
(UM CIGANO ENTROU NA CASA)
(UM CIGANO FOI VISTO COM O COLAR) )
( (A CAMAREIRA GANHAVA POUCO)
SE
(O SALARIO DA CAMAREIRA ERA BAIXO)
(A UNICA FONTE DE RENDA DA CAMAREIRA
ERA O SALARIO) )
( (UM CIGANO ENTROU NA CASA)
SE
(HAVIA RASTROS DE BOTAS NO JARDIM)
(CIGANOS SEMPRE USAM BOTAS)
(APENAS CIGANOS USAM BOTAS) ) )
```

LISTA DE HIPOTHESES

```
( (A CAMAREIRA ROUBOU O COLAR)
(OS CIGANOS ROUBARAM O COLAR)
(A GRALHA ROUBOU O COLAR) )
```

tecedentes. Quando o conseqüente de uma regra é igual a uma hipótese, a regra é dita relevante para a hipótese.

O algoritmo usado para estabelecer se uma hipótese é verdadeira ou falsa é denominado **VERIFIQUE** e está descrito no box anexo. Nós o implementamos em BASIC(Listagem 2), LISP(Listagem 3) e LOGO(Listagem 4). LOGO é um dialeto do LISP usado para ensinar técnicas básicas de Inteligência Artificial a crianças. Podemos dizer que LOGO e LISP são praticamente iguais.

COMO USAR OS PROGRAMAS

A primeira coisa a fazer é digitar o programa que estiver na linguagem de sua preferência. Se você for usar o programa em LISP no Apple II, digite antes a função DEF. Se você usar o programa em LISP num TRS-80 como o intérprete da Supersoft, não tecle a definição de DEF.

Listagem 2 - BASIC

```
10 REM SISTEMA ESPECIALISTA
20 REM LINGUAGEM: BASIC
30 REM COMPUTADOR: APPLE II
40 REM AUTOR: ANTONIO COSTA.
50 REM
60 REM *****
70 REM DESCRICAO DE VARIAVEIS
80 REM *****
90 REM
100 REM FR$: FRASES.
110 REM HIP: PONTEIROS PARA HIPO
    TESES ARMazenADAS EM FR$
120 REM FT: PONTEIROS PARA FATOS
    ARMazenADOS EM FR$
130 REM CNSQ: PONTEIROS PARA OS
    CONSEQUENTES ARMazenADOS EM
    FR$.
140 REM AN: NUMERO DE ANTECEDENT
    ES DA REGRA CUJO CONSEQUENTE
    EH APONTADO POR FR$.
150 REM OS ANTECEDENTES SAO POS
    TOS EM FR$ LOGO APOS OS CONS
    EQUENTES.
160 REM RL: PONTEIROS PARA AS RE
    GRAS RELEVANTES.
170 REM ST: FILHA PARA SIMULAR R
    ECURSIVIDADE.
180 REM IR: NUMERO DE FRASES.
190 REM IH: NUMERO DE HIPOTHESES.
200 REM IT: NUMERO DE FATOS.
210 REM RG: NUMERO DE REGRAS.
220 REM RS: NUMERO DE REGRAS REL
    EVANTES.
930 REM *****
940 REM == ESPECIALISTA ====
1000 REM
1010 DIM FR$(100),HIP(20)
1015 DIM FT(100),CNSQ(50)
1020 DIM RL(50),ST(100)
1025 DIM AN(50)
1030 IR = 0: IH = 0: IT = 0
1035 RG = 0: RS = 0
1040 HOME
1050 PRINT " ** SISTEMAS "
1055 PRINT "ESPECIALISTAS **"
1060 PRINT "ESCOLHA UMA DAS "
1065 PRINT "POSSIBILIDADES "
1067 PRINT "ABAIXO"
1068 PRINT : PRINT
1070 PRINT " A - INICIALIZAR"
1080 PRINT " B - LER FATOS DO T
    ECLADO"
1085 PRINT " LEITURA DE FAT
    OS E' OPTATIVA"
1110 PRINT " C - LER HIPOTHESES
    DO TECLADO"
1120 PRINT " D - LER REGRAS DO
    TECLADO"
1150 PRINT " E - LER REGRAS DO
    DISCO"
1160 PRINT " F - GRAVAR REGRAS"
1170 PRINT " G - INVESTIGAR HIP
    OTESES"
1172 PRINT " H - ESVAZIAR LISTA
    DE FATOS"
1173 PRINT " I - ESVAZIAR LISTA
    DE HIPOTHESES"
1174 PRINT " J - ADICIONAR NOVA
    S REGRAS"
1176 PRINT " K - IMPRIMIR FATOS"
1177 PRINT " L - ABANDONAR PROG
    RAMA"
1180 GET A$
1190 OP = ASC (A$) - 64
1200 IF (OP < 1) OR (OP > 12) THEN
    1180
1210 ON OP GOSUB 2000,6500,6000,
    7000,2800,2600,1240,12000,12
    100,7020,12200
1212 IF OP < 12 THEN 1040
1215 REM *****
1220 END
1230 REM *****
1232 REM *****
1234 REM == PROVEUMA =====
1240 IF IH = 0 THEN PRINT "NAO
    TENHO HIPOTHESES PARA TRABALH
    AR.": PRINT "VOCE NAO PODERIA
    A ME FORNECER ALGUMAS?": GOSUB
    6000
1250 IF RG = 0 THEN PRINT "NAO
    TENHO REGRAS PARA TRABALHAR.
    ": PRINT "VOCE NAO PODERIA M
    E FORNECER ALGUMAS?": GOSUB
    7000
1260 FOR IJ = 1 TO IH
1270 H = HIP(IJ): GOSUB 5000
1280 IF T2 = 1 THEN IJ = IH
1290 NEXT IJ
1300 IF T2 = 0 THEN PRINT "NENH
    UMA HIPOTHESE PODE SER VERIFI
    CADA"
1310 IF T2 = 1 THEN PRINT "A HI
    POTESE ";FR$(H);" E' VERDADE
    IRA"
1320 PRINT "APORTE QUALQUER TECL
    A PARA CONTINUAR"
1330 GET A$
1340 RETURN
1350 REM *****
1900 REM *****
1950 REM ROTINA DE INICIALIZAR
2000 IR = 0: IH = 0: IT = 0: RG = 0:
    RS = 0: PT = 0
2010 RETURN
2015 REM *****
2580 REM *****
2590 REM == GRAVA REGRAS =====
2600 D$ = CHR$(4): REM CTRL-D
2601 PRINT D$
2605 PRINT D$;"OPEN REGRAS"
2610 PRINT D$;"WRITE REGRAS"
2615 PRINT RG: PRINT IR
2620 FOR I = 0 TO IR
2625 PRINT FR$(I)
2630 NEXT I
2635 FOR I = 0 TO RG
2640 PRINT CNSQ(I): PRINT AN(I)
2645 NEXT I
2650 PRINT D$;"CLOSE REGRAS"
2665 RETURN
2670 REM *****
2790 REM *****
2795 REM == LER REGRAS DO DISCO
2800 D$ = CHR$(4)
2805 PRINT D$;"OPEN REGRAS"
2810 PRINT D$;"READ REGRAS"
2815 INPUT RG: INPUT IR
2820 FOR I = 0 TO IR
2825 INPUT FR$(I)
2830 NEXT I
2835 FOR I = 0 TO RG
2840 INPUT CNSQ(I): INPUT AN(I)
2845 NEXT I
2855 PRINT D$;"CLOSE REGRAS"
2860 RETURN
2870 REM *****
2900 REM *****
3000 REM == DESCREVE RACIOCINIO
3010 PRINT "USANDO A REGRA:"
3020 PRINT FR$(CNSQ(RL(J))): SE
    "
3030 FOR K = 1 TO NA
3040 PRINT " ";FR$(CNSQ(RL(J)) +
    K);
    IF (K < NA) THEN PRINT " E
    "
3060 NEXT K
3070 PRINT
3080 PRINT "CONSEGUI PROVAR QUE
    ";FR$(CNSQ(RL(J)))
3090 RETURN
3100 REM *****
3900 REM *****
4000 REM == INVESTIGUE =====
4010 PRINT "A HIPOTHESE ABAIXO E'
    VERDADEIRA? (S/N)"
4020 F$ = FR$(H)
4030 PRINT F$
4040 INPUT RESP$
4050 IF RESP$ = "S" THEN GOSUB
    9000:T2 = 1: RETURN
4060 IF RESP$ = "N" THEN T2 = 0:
    GOSUB 8000: GOSUB 9000: RETURN
4070 PRINT "POR FAVOR, RESPONDA
    S OU N"
4080 GOTO 4000
4100 REM *****
4900 REM *****
4910 REM == VERIFIQUE =====
5000 REM VERIFICA SE HIPOTHESE E
    VERDADEIRA: ARGUMENTO EM H
5010 REM RESPOSTA EM T2
5020 F$ = FR$(H): GOSUB 8500
5030 IF T1 = 1 THEN T2 = 1: GOTO
    5330
5040 GOSUB 8000: GOSUB 8500
```


PROGRAMAS QUE RACIOCINAM

(continuação Listagem 2 - BASIC)

```
5050 IF T1 = 1 THEN T2 = 0: GOTO 5330
5060 REM
5070 GOSUB 7500
5080 REM
5090 IF RS = 0 THEN GOSUB 4000: GOTO 5330
5095 J = 1
5100 REM
5110 NA = AN(RL(J))
5115 K = 1
5120 REM
5130 PT = PT + 1: ST(PT) = H: PT = PT + 1: ST(PT) = J
5140 PT = PT + 1: ST(PT) = K: PT = PT + 1: ST(PT) = NA
5150 FOR J1 = 0 TO RS
5160 PT = PT + 1: ST(PT) = RL(J1)
5170 NEXT J1
5180 PT = PT + 1: ST(PT) = RS
5190 H = CNSQ(RL(J)) + K: GOTO 5000
5200 RS = ST(PT): PT = PT - 1
5210 FOR J1 = RS TO 0 STEP - 1
5220 RL(J1) = ST(PT): PT = PT - 1
5230 NEXT J1
5240 NA = ST(PT): PT = PT - 1
5250 K = ST(PT): PT = PT - 1: J = ST(PT): PT = PT - 1
5260 H = ST(PT): PT = PT - 1
5270 IF T2 = 0 THEN K = NA
5280 K = K + 1: IF K < = NA GOTO 5120
5290 IF T2 = 1 THEN GOSUB 3000: J = RS
5300 J = J + 1: IF J < = RS THEN 5100
5330 IF PT > 1 THEN 5200
5340 RETURN
5400 REM *****
5500 REM *****
6000 REM LER HIPOTHESES
6005 IH = 0
6010 PRINT "QUAL E' A HIPOTHESE?" : INPUT H$
6030 IH = IH + 1: IR = IR + 1: FR$(IR) = H$: HIP(IH) = IR
6040 PRINT "VOCE QUER LER OUTRA HIPOTHESE? (S/N)"
6050 GET RESP$: IF (RESP$ < > "S") AND (RESP$ < > "N") THEN 6050
6060 IF RESP$ = "S" THEN 6010
6065 RETURN
```

```
6070 REM *****
6490 REM *****
6500 REM == LER FATOS ==
6510 PRINT "QUAL E' O FATO?": INPUT F$
6530 IT = IT + 1: IR = IR + 1: FR$(IR) = F$: FT(IT) = IR
6540 PRINT "VOCE QUER LER OUTRO FATO? (S/N)"
6550 GET RESP$
6555 IF (RESP$ < > "S") AND (RESP$ < > "N") THEN GOTO 6550
6560 IF RESP$ = "S" THEN 6510
6565 RETURN
6570 REM *****
7000 REM == LER REGRAS ==
7005 PRINT
7010 RS = 0
7020 PRINT "TODAS AS REGRAS DEVE M TERMINAR COM ."
7030 PRINT "TECLE A REGRA.": INPUT BU$
7040 GOSUB 11000
7050 RG = RG + 1: IR = IR + 1: FR$(IR) = FS$: CNSQ(RG) = IR
7060 NA = 0
7070 GOSUB 11000
7080 IF FS$ = "" THEN INPUT "CONT. DA REGRA ": BU$: GOTO 7070
7090 IF FS$ < > "." THEN NA = NA + 1: IR = IR + 1: FR$(IR) = FS$: GOTO 7070
7100 AN(RG) = NA
7110 PRINT "VOCE QUER LER OUTRA REGRA? (S/N)"
7120 GET RESP$
7125 IF (RESP$ < > "S") AND (RESP$ < > "N") THEN GOTO 7120
7130 IF RESP$ = "S" THEN GOTO 7030
7135 RETURN
7200 REM *****
7490 REM *****
7500 REM == RELEVANTES ==
7510 RS = 0
7520 FOR I = 0 TO RG
7530 IF FR$(CNSQ(I)) = FR$(H) THEN RS = RS + 1: RL(RS) = I
7540 NEXT I
7550 RETURN
```

```
7560 REM *****
7570 REM *****
8000 REM == NEG ==
8010 REM PARAMETRO EM F$
8020 IF LEFT$(F$,4) = "NAD" THEN F$ = RIGHT$(F$, LEN(F$) - 4): RETURN
8030 F$ = "NAD" + F$: RETURN
8100 REM *****
8490 REM *****
8495 REM == MEMBER ==
8500 REM MEMBER VERIFICA SE F$ ESTA NA LISTA DE FATOS. RE SPOSTA EM T1.
8510 T1 = 0
8520 FOR I1 = 1 TO IT
8530 IF FR$(FT(I1)) = F$ THEN I1 = I1: T1 = 1
8540 NEXT I1
8550 RETURN
8600 REM *****
8700 REM *****
9000 REM GUARDA FATO. RESPOSTA EM T1. ARGUMENTO EM F$
9010 GOSUB 8500
9020 IF T1 = 1 THEN RETURN
9030 REM
9040 REM
9100 IR = IR + 1: IT = IT + 1
9110 FR$(IR) = F$: FT(IT) = IR
9120 RETURN
9200 REM *****
9210 REM *****
10000 REM == ANALISADOR LEXICO
10010 WR$ = ""
10020 IF LEN(BU$) = 0 THEN RETURN
10030 CH$ = LEFT$(BU$,1)
10035 IF LEN(BU$) > 1 THEN BU$ = RIGHT$(BU$, LEN(BU$) - 1): GOTO 10040
10036 BU$ = ""
10040 IF (CH$ = ".") AND (LEN(BU$) > 0) THEN GOTO 10030
10050 IF ASC(CH$) < 33 THEN RETURN
10060 IF LEN(BU$) = 0 THEN WR$ = CH$: RETURN
10070 WR$ = CH$
10080 IF WR$ = "." THEN RETURN
10090 CH$ = LEFT$(BU$,1)
```

Listagem 3 - LISP

```
(DEFINE (DEF (FLAMBDA (S)
  (PUT (CAR S) 'EXPR (CONS 'LAMBDA (CDR S))) )))
```

```
(DEF ESPECIALISTA (HIPOS)
  (COND ((NULL HIPOS) NIL)
        ((VERIFIQUE (CAR HIPOS)) (CAR HIPOS))
        (ESPECIALISTA (CDR HIPOS))))
```

```
(DEF VERIFIQUE (HIPO)
  (COND ((MEMBER HIPO FATOS) T)
        ((MEMBER (NEG HIPO) FATOS) NIL)
        ((PROVEUMA (RELEVANTES HIPO REGRAS)) T)
        ((PRINT) (PRINT ' (INVESTIGUE SE)) (PRINT HIPO)
         (INVESTIGUE)))
```

```
(DEF RELEVANTES (HIP REGRS)
  (COND ((NULL REGRS) NIL)
        ((EQUAL HIP (CAR (CAR REGRS)))
         (CONS (CDR (CDR (CAR REGRS)))
                (RELEVANTES HIP (CDR REGRS))))
        (T (RELEVANTES HIP (CDR REGRS)))))
```

```
(DEF PROVEUMA (RLS)
  (COND ((NULL RLS) NIL)
        ((VERDADEIRAS (CAR RLS)) T)
```

```
((PROVEUMA (CDR RLS)))
```

```
(DEF VERDADEIRAS (ANTEC)
  (COND ((NULL ANTEC) T)
        ((VERIFIQUE (CAR ANTEC)) (VERDADEIRAS (CDR ANTEC)))))
```

```
(DEF INVESTIGUE ()
  (PROG (VF)
    (PRINT (CHR 7))
    (SETQ VF (GETCHR))
    (COND ((EQUAL VF 'V) (PRINT 'VERDADEIRA)
           (SETQ FATOS (CONS HIPO FATOS)) (RETURN T))
          ((EQUAL VF 'F) (PRINT 'FALSA)
           (SETQ FATOS (CONS (NEG HIPOS) FATOS)) (RETURN NIL))
          (T (PRINT ' (RESPONDA APENAS COM V/F)) (INVESTIGUE)))))
```

```
(DEF NEG (H)
  (COND ((EQUAL (CAR H) 'NEGACAO-DE:) (CDR H))
        (T (CONS 'NEGACAO-DE: H))))
```

```
(DEF MEMBER (E L)
  (COND ((NULL L) NIL)
        ((EQUAL E (CAR L)) T)
        (T (MEMBER E (CDR L)))))
```

```
(DEF LISTAFATOS (FATOS)
  (COND ((NULL FATOS)
        ((PRINT (CAR FATOS))
         (LISTAFATOS (CDR FATOS)))))
```

Listagem 4 - LOGO

```

10092 IF LEN (BU$) > 1 THEN BU$
      = RIGHT$ (BU$, LEN (BU$) -
1): GOTO 10100
10100 IF (ASC (CH$) < 33) THEN
      RETURN
10105 IF CH$ = "." THEN BU$ = CH
      $ + BU$: RETURN
10110 WR$ = WR$ + CH$
10120 IF LEN (BU$) = 0 THEN RETURN

10130 GOTO 10080
10200 REM *****
10205 REM *****
10210 REM ANALISADOR SINTATICO
11000 FS$ = ""
11010 GOSUB 10000
11015 IF WR$ = "SE" OR WR$ = "E"
      THEN GOTO 11010
11020 IF WR$ = "" THEN RETURN
11025 FS$ = WR$
11030 IF WR$ = "." THEN RETURN

11040 GOSUB 10000
11050 IF WR$ = "" OR WR$ = "SE" OR
      WR$ = "E" THEN RETURN
11060 IF WR$ = "." THEN BU$ = ".",
      " + BU$: RETURN
11070 FS$ = FS$ + " " + WR$
11080 GOTO 11040
11100 REM *****

11110 REM *****
12000 REM ESVAZIA LISTA DE FATO
      S
12010 IT = 0
12020 RETURN
12030 REM *****

12040 REM *****
12100 REM ESVAZIA LISTA DE HIPO
      TESES
12110 IH = 0
12115 RETURN
12150 REM *****

12200 REM *****
12210 REM === LISTAFATOS =====
12220 FOR I = 1 TO IT
12230 PRINT FR$(FT(I))
12240 NEXT I
12245 PRINT "APORTE QUALQUER TEC
      LA PARA CONTINUAR"
12246 GET A$
12250 RETURN
12260 REM *****

```

O programa BASIC pode ser usado sem modificações no Apple e, com pequenas modificações, no TRS-80 e no Sinclair. A primeira destas modificações é substituir os comandos GET por comandos INKEY\$. Assim, o comando GET A\$ da linha 1180 deve ser substituído por

```

1180 A$ = INKEY$; IF
A$ = "" THEN 1180

```

As sub-rotinas 2600 e 2800 são usadas, respectivamente, para gravar regras em disco e ler regras do disco. O que a sub-rotina 2600 faz é simplesmente gravar as variáveis RG e IR e as matrizes FR\$, CNSQ e AN no disco. Já a rotina 2800 recupera estas mesmas matrizes e variáveis. Ora, comandos de leitura e gravação variam enormemente de computador para computador. Isto significa que, se seu computador não for Apple, você deve reescrever estas duas sub-rotinas.

Após ter digitado o programa, você deve introduzir as hipóteses e regras no computador. Se você estiver usando LISP, tecle (SETQ HIPOTeseS (READ)) e em seguida tecle as três hipóteses da *Listagem 1*. Se você estiver usando LOGO, tecle MAKE "HIPOTeseS e em segui-

```

TO ESPECIALISTA :HIPOTeseS
IF :HIPOTeseS = [] OUTPUT "HIPOTeseS FALSAS"
IF VERIFIQUE FIRST :HIPOTeseS OUTPUT FIRST :HIPOTeseS
OUTPUT ESPECIALISTA BUTFIRST :HIPOTeseS
END

```

```

TO VERIFIQUE :HIPO
IF MEMBER :HIPO :FATOS OUTPUT "TRUE"
IF MEMBER ( NEG :HIPO ) :FATOS OUTPUT "FALSE"
IF PROVEUMA ( RELEVANTES :HIPO :REGRAS ) OUTPUT "TRUE"
CLEARINPUT PRINT "
( PRINT "INVESTIGUE SE" :HIPO )
OUTPUT INVESTIGUE :HIPO
END

```

```

TO RELEVANTES :HIP :REGS
IF :REGS = [] OUTPUT []
IF :HIP = FIRST FIRST :REGS OR FPUT BF BF FIRST :REGS RELEVANTES :HIP BF :REGS
OUTPUT RELEVANTES :HIP BUTFIRST :REGS
END

```

```

TO PROVEUMA :RLS
IF :RLS = [] OUTPUT "FALSE"
IF VERDADEIRAS FIRST :RLS OUTPUT "TRUE"
OUTPUT PROVEUMA BUTFIRST :RLS
END

```

```

TO VERDADEIRAS :HIPS
IF :HIPS = [] OUTPUT "TRUE"
IF VERIFIQUE FIRST :HIPS OUTPUT VERDADEIRAS? BUTFIRST :HIPS
OUTPUT "FALSE"
END

```

```

TO INVESTIGUE :HIP
PRINT1 CHAR 7
MAKE "VF READCHARACTER
IF :VF = "V PRINT "VERDADEIRA MAKE "FATOS FPUT :HIP :FATOS OUTPUT "TRUE"
IF :VF = "F PRINT "FALSA MAKE "FATOS FPUT ( NEG :HIP ) :FATOS OUTPUT "FALSE"
PRINT1 [RESPONDA APENAS COM V/F:]
OUTPUT INVESTIGUE :HIP
END

```

```

TO NEG :H
IF FIRST :H = "NEGACAO-DE: OUTPUT BUTFIRST :H
OUTPUT SENTENCE "NEGACAO-DE: :H
END

```

```

TO MEMBER :E :L
IF :L = [] OUTPUT "FALSE"
IF :E = FIRST :L OUTPUT "TRUE"
OUTPUT MEMBER :E BUTFIRST :L
END

```

```

TO LISTAFATOS :FATOS
IF :FATOS = [] STOP
PRINT FIRST :FATOS
LISTAFATOS BUTFIRST :FATOS
END

```

da entre com as hipóteses da *Listagem 1*, substituindo os parênteses redondos por parênteses quadrados. A introdução das regras é análoga: em LISP, tecla (SETQ REGRAS (READ)), seguido das regras da *Listagem 1*; em LOGO, tecla MAKE "REGRA, seguido das regras da *Listagem 1*.

O programa BASIC é dirigido por um menu. Tecla RUN e o menu ser-lhe-á apresentado. Escolha a opção C e o programa lhe pedirá uma hipótese. Tecla a primeira das hipóteses da *Listagem 1* sem os parênteses e sem qualquer marca de pontuação. O programa perguntará se você quer ler outra hipótese. Responda S e tecla a hipótese seguinte. Após teclar as três hipóteses, pressione N e o menu será recolocado na tela. Escolha a opção D e o programa pedirá uma regra. Comece a entrar com as regras da *Listagem 1* sem os parênteses e obedecendo às seguintes restrições:

- 1 — o *consequente* deve ser separado dos *antecedentes* por SE;
 - 2 — os *antecedentes* devem ser separados um do outro por E;
 - 3 — toda regra deve terminar com ponto.
- A terceira regra, por exemplo, deve ser teclada assim:

A GRALHA ROUBOU O
COLAR SE A JANELA
ESTAVA ABERTA E O
COLAR ERA
BRILHANTE E
GRALHAS GOSTAM DE
COISAS BRILHANTES.

Para que o programa BASIC comece a investigação, basta escolher a opção G do menu. Para iniciar a investigação em LISP, tecla (ESPECIALISTA HIPOTESSES). Em LOGO, a investigação terá início quando você teclar (ESPECIALISTA :HIPOTESSES).

O algoritmo VERIFIQUE

. Passo 1 — Examine se a hipótese está na lista de fatos. Se estiver, ela é verdadeira pois, como disse Aristóteles, contra fatos não há argumentos.

. Passo 2 — Examine se a negação da hipótese está na lista de fatos. Se estiver, a hipótese é falsa.

. Passo 3 — Construa a lista de todas as regras cujos consequentes sejam iguais à hipótese. Estas regras, conforme vimos, são relevantes para demonstrar a veracidade de uma hipótese.

. Passo 4 — Entre as regras relevantes, tente encontrar uma cujos antecedentes sejam todos verdadeiros. Se você conseguir, a hipótese é verdadeira. E como demonstrar que todos os antecedentes de uma regra relevante são verdadeiros? Simplesmente usando o algoritmo VERIFIQUE!

. Passo 5 — Se não houver nenhuma regra relevante não será possível concluir pela veracidade ou falsidade da hipótese usando processos dedutivos. Neste caso, o sistema especialista deve pedir ao usuário que investigue se a hipótese é verdadeira ou falsa. Aqui ele se assemelha ao detetive Nero Wolf que, por ser muito gordo, não podia sair de casa e pedia a seu assistente que investigasse tudo quanto não era possível deduzir.

APLICAÇÕES PARA O PROGRAMA DETETIVE

As hipóteses e regras dadas na *Listagem 1* foram escolhidas para que o programa resolva o mistério apresentado em *La gazza ladra*. A história da *gazza ladra* foi aproveitada pelo músico Joaquim Rossini para compor uma ópera e pelo desenhista Hergé para a aventura de Tintin denominada *As jóias da Castafiore*. Na aventura de Tintin, ciganos que estavam acampados nas proximidades da mansão do capitão Haddock são acusados injustamente de roubarem o valioso colar da soprano Castafiore. O herói Tintin, porém, usando raciocínio regressivo,

descobre que uma gralha, atraída pelo brilho da jóia, entrou no quarto e levou o colar para seu ninho.

O leitor, substituindo as hipóteses e regras da *Listagem 1*, poderá fazer seu computador resolver outros mistérios. Assim, o computador poderá representar o papel de Guilherme de Baskerville em *O nome da rosa* ou de Hercule Poirot em *O assassinato no Expresso Oriental*. Para que a máquina tenha êxito, entretanto, é necessário escolher as regras rigorosamente.

A escolha cuidadosa das regras é tão importante que surgiu nos Estados Unidos um tipo de profissional, o engenhei-

Listagem 5 - Exemplo de execução

A HIPOTESE ABAIXO E' VERDADEIRA? (S/N)
CIGANDOS SEMPRE USAM BOTAS
?S
A HIPOTESE ABAIXO E' VERDADEIRA? (S/N)
APENAS CIGANDOS USAM BOTAS
?N
A HIPOTESE ABAIXO E' VERDADEIRA? (S/N)
A CAMAREIRA GASTAVA MUITO
?S
A HIPOTESE ABAIXO E' VERDADEIRA? (S/N)
O SALARIO DA CAMAREIRA ERA BAIXO
?S
A HIPOTESE ABAIXO E' VERDADEIRA? (S/N)
A UNICA FONTE DE RENDA DA CAMAREIRA ERA O SALARIO
?S
USANDO A REGRA:
A CAMAREIRA GANHAVA POUCO SE
O SALARIO DA CAMAREIRA ERA BAIXO
E A UNICA FONTE DE RENDA DA CAMAREIRA ERA O SALARIO

CONSEGUI PROVAR QUE A CAMAREIRA GANHAVA POUCO
USANDO A REGRA:
A CAMAREIRA PRECISAVA DE DINHEIRO SE
A CAMAREIRA GASTAVA MUITO
E A CAMAREIRA GANHAVA POUCO

CONSEGUI PROVAR QUE A CAMAREIRA PRECISAVA DE DINHEIRO
A HIPOTESE ABAIXO E' VERDADEIRA? (S/N)
ERA FACIL VENDER O COLAR
?S
USANDO A REGRA:
A CAMAREIRA ROUBOU O COLAR SE
A CAMAREIRA PRECISAVA DE DINHEIRO
E ERA FACIL VENDER O COLAR

CONSEGUI PROVAR QUE A CAMAREIRA ROUBOU O COLAR
A HIPOTESE A CAMAREIRA ROUBOU O COLAR E' VERDADEIRA
APORTE QUALQUER TECLA PARA CONTINUAR

ro de conhecimento, cuja única função é escolher as regras. O engenheiro de conhecimentos entrevista médicos, jogadores, geólogos ou químicos e tenta transformar a experiência em regras que serão usadas por programas especialistas. A seguir fornecemos algumas dicas de como escolher as regras.

Para tornar a investigação emocionante, é preciso criar regras que usem outras regras. Regras para cuja comprovação é necessário usar outras regras forçam o computador a um raciocínio indireto e isto torna o processo de dedução muito interessante. Tentemos esclarecer este ponto com um exemplo. Se quero fornecer ao computador meios de provar que (A CAMAREIRA ROUBOU O COLAR), posso simplesmente colocar a regra:

((A CAMAREIRA ROUBOU O COLAR) SE (A CAMAREIRA PRECISAVA DE DINHEIRO)).

A investigação, porém, tornar-se-á muito mais interessante se fornecermos uma regra que permita deduzir que (A CAMAREIRA PRECISAVA DE DINHEIRO). Esta regra poderia ser:

((A CAMAREIRA PRECISA DE DINHEIRO) SE (A CAMAREIRA GASTAVA MUITO))

E a emoção aumentará se fornecermos ao computador a seguinte regra para provar que (A CAMAREIRA GASTAVA MUITO):

((A CAMAREIRA GASTAVA MUITO) SE (A CAMAREIRA COMPRAVA OBJETOS CAROS)).

Na listagem 5 temos um exemplo de execução do nosso programa detetive.

As funções do programa especialista

. A função **ESPECIALISTA** recebe uma lista de hipóteses e, usando a função **VERIFIQUE**, tenta encontrar uma que seja verdadeira.

. A função **VERIFIQUE** recebe uma hipótese e, fazendo uso dos cinco passos já descritos, descobre se ela é verdadeira ou falsa.

. A função **RELEVANTES** seleciona dentro da lista de regras todas as regras relevantes da hipótese que recebeu de **VERIFIQUE**.

. A função **PROVEUMA** executa o passo 4, isto é, tenta provar que uma das regras relevantes é verdadeira.

. A função **VERDADEIRAS** é a mais simples e interessante. Ela entrega para a função **VERIFIQUE** uma antecedente para ser provada (da mesma forma que **ESPECIALISTA** entrega uma hipótese). Note que **VERDADEIRAS** faz uso de todo programa e, portanto, de si mesma. No programa **BASIC**, **VERDADEIRAS** foi incorporada a **VERIFIQUE**.

. A função **INVESTIGUE** executa o último passo, a saber, recorre ao usuário para decidir se a hipótese é verdadeira ou falsa.

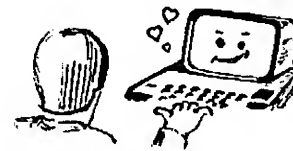
. A função **NEG** recebe uma hipótese ou um antecedente e retorna sua negação.

. A função **MEMBER** verifica se uma hipótese está na lista de fatos.

Antonio Eduardo Costa Pereira é formado em Engenharia Eletrônica pela Escola Politécnica da USP e em Física pelo Instituto de Física da USP. Fez mestrado em Ciência Espacial no Instituto de Pesquisas Espaciais em São José dos Campos, SP, e doutorado em Engenharia Eletrônica na Cornell University em Ithaca, Nova Iorque (EUA). Atualmente é professor na USP.

André Gurgel é analista de sistemas da sucursal de São Paulo da Cia. de Seguros Aliança da Bahia e graduando do curso de Biologia da Universidade de São Paulo.

você e seu micro



Um relacionamento assim merece um programa exclusivo

- A Informatic Service desenvolve programas conforme suas necessidades.
- Presta-lhe total assistência através de uma Assessoria de Sistemas exclusiva.
- Promove a relação Homem-Software-Máquina de forma perfeita

Consulte-nos ainda hoje.

INFORMATIC SERVICE DO BRASIL

Av. 13 de Maio, 47 - grupo 2707

Fone: (021) 262-8769 - RJ

repro



**CIÊNCIA MODERNA
COMPUTAÇÃO**

A Única Especializada em Livros
e Revistas p/ Microcomputação

Publicações p/ IBM PC/XT e Similares:

• IBM PC EXPANSION AND SOFTWARE GUIDE (Due) 3/E	41.527,00
• SPREADSHEET SOFTWARE FROM VISICALC TO 1.2.3. (Due)	39.077,00
• GRAPHICS FOR THE IBM PC (Koriles) acompanha diskette	116.150,00
• 100 READY-TO-RUN PROGRAMS AND SUBROUTINES FOR THE IBM PC (Breitz)	40.425,00
• BUSINESS APPLICATIONS FOR IBM PERSONAL COMPUTER (Zimmerman)	43.977,00
• HOW TO GET STARTED WITH MS-DOS (Townsend)	20.227,00
• USEFUL BASIC PROGRAMS FOR THE IBM PC (Trost)	12.977,00
• IBM BASIC (Payne)	23.127,00
• IBM'S PERSONAL COMPUTER 2ND EDITION INCLUDES XT (Due)	39.077,00
• ADVANCED BASIC AND BEYOND FOR THE IBM PC (Goldstein)	48.877,00
• SOFTWARE SOLUTIONS FOR THE IBM PC. A Practical Guide to Dbase II Lotus 1.2.3 Visicalc etc. (Willmott)	36.627,00
• IBM DATA FILES: A Basic Tutorial (Miller)	23.125,00
• HANDBOOK OF BASIC FOR THE IBM PC (Schneider)	48.877,00
• PC DOS USING THE IBM PC OPERATING SYSTEM (Ashley)	36.627,00
• IBM PC DATA FILE PROGRAMMING (Brown)	36.627,00

Visite-nos ou peça uma lista do nosso estoque específica para seu micro.

Atendemos também por Reemb. Postal e Varig.

Av. Rio Branco, 156 — loja 127 — subsolo
Rio de Janeiro — Tels. 262-5723/240-9327

Corrigindo provas no micro

Lawrence Falconer King

Este programa corrige e calcula as notas de até duzentos alunos por turma, de provas também de até duzentas questões. Para cada turma corrigida é criado um arquivo em disco ou fita, onde são gravados os nomes dos alunos e suas respectivas notas. Estes arquivos poderão ser listados no vídeo ou na impressora.

O programa se apresenta dividido em três partes. A primeira envolve a montagem do gabarito de respostas, a correção das provas dos alunos e a criação do arquivo por turma. A segunda parte é relativa à listagem dos arquivos existentes em disco ou fita. A última possibilita a transferência de arquivos entre discos, fitas, e entre disco e fita (e vice-versa).

Como este programa ocupa uma grande área de memória, somente microcomputadores com 48K RAM é que poderão suportá-lo. Para sistemas operacionais diferentes do DOS-500 e do TRSDOS (do TRS-80), algumas modificações poderão ser necessárias nas rotinas relativas à leitura-gravação de arquivos em disco. Para isto, sugiro que você consulte seu manual referente ao sistema operacional.

Para carregar o programa, utilizamos o comando RUN ou CLOAD. Após isso, indicamos a tarefa que desejamos realizar, isto é, criar, listar ou transferir. Devemos teclar uma das três letras C, L ou T, de acordo com o trabalho a ser efetuado, ou então teclamos FIM para terminar o processamento, se for o caso.

Correção de Provas

```
10 REM      =====
20 REM      =  PROGRAMA DE CORRECAO DE PROVAS =
30 REM      =  AUTOR: LAWRENCE FALCONER KING =
40 REM      =  TF: (0242) 42-1919 =
50 REM      =  PETROPOLIS/RJ =
60 REM      =====
80 CLEAR 10000
85 DIM M$(200,4), IX(200)
90 M1$="CLASSIFICAR POR (A)LUNO OU POR (N)OTA?"
100 A$=CHR$(34)
110 G$=" "
140 CLS:PRINT TAB(20)"PROGRAMA DE CORRECAO DE PROVAS"
150 INPUT "(C)RIAR, (L)ISTAR OU (T)RANSFERIR";T$:IF T$="FIM" THEN CLS:
    END ELSE T$=LEFT$(T$,1)
160 IF T$="L" GOTO 1080 ELSE IF T$="C" GOTO 200 ELSE IF T$="T" GOTO 116
0 ELSE GOTO 150
170 REM
175 REM      =====
180 REM      ROTINA DE CORRECAO E CRIACAO DOS ARQUIVOS
185 REM      =====
190 REM
200 REM      PECA DADOS DE ENTRADA
210 REM
220 REM
230 GOSUB 2500
240 REM
250 REM      MONTAGEM DO GABARITO
260 REM
265 H1$="GABARITO DA DISCIPLINA DE ":H2$=M$(I,1):H3$="QUESTAO":H4$="RESPOSTA"
":X=0
270 IF G$="F" CLS:FOR I=1 TO NQ:M$(I,1)="0":NEXT I:PRINT@10,"OUTRO GABAR
ITO (S/N) ?":INPUT R$:IF LEFT$(R$,1)="N" GOTO 500
280 CLS
290 PRINT@ 13-INT(LEN(NM$)/2), "MONTAGEM DO GABARITO DA DISCIPLINA DE ";
NM$
300 I=1
310 PRINT@173,"      ":PRINT@ 146, "ENTRE RESPOSTA DA QUESTAO";I;
320 INPUT M$(I,0):IF LEN(M$(I,0))=0 GOTO 310
330 IF M$(I,0)(">"FIM" THEN GOTO340 ELSE GOTO 390
```

PARA (C)RIAR

Se teclamos C, o programa entra na rotina de criação de arquivo, que, por sua vez, é dividida em várias sub rotinas: entrada de dados, montagem do gabarito, correção das provas dos alunos, classificação da turma por aluno ou por nota, gravação da turma em arquivo de disco ou fita e índice de aproveitamento das questões.

Na primeira sub-rotina, devemos entrar com vários dados necessários ao processamento. Começamos, então, com o nome da turma, que deve ser fornecido como um nome de arquivo válido para o DOS, caso as unidades de disco sejam utilizadas.

Vejamos, como exemplo, estes dois tipos de nome de turma: **ISERIE** e **1SERIE/MAT**. No primeiro caso, fica especificado apenas o nome da turma e pode ser utilizado se o arquivo de saída é por fita ou disco, sendo que, neste último, em caso de não haver outra disciplina para a mesma turma. No segundo exemplo, temos o nome da turma (até o máximo de sete caracteres) seguido de uma extensão composta de "/" e mais três letras (ou algarismos, no máximo três caracteres), o que serve para identificar o arquivo de notas de matemática da turma **ISERIE**, dos arquivos de notas de outras disciplinas desta mesma turma.

Em relação aos dois exemplos acima, é mais conveniente que utilizemos sempre o segundo, isto é, com a extensão (mesmo para arquivos em fita) e que esta extensão seja uma abreviatura da disciplina. O fato de o nome da turma iniciar por um algarismo é irrelevante para o programa, pois quando é criado o arquivo em disco, a letra T é prefixada ao nome, de modo que, ao se pedir o diretório do disco, após a criação do arquivo, o nome será **T1SERIE/MAT** (usando o exemplo acima).

Entramos, agora, com o nome da disciplina cujas provas serão corrigidas. Este nome deve ser teclado corretamente, pois será gravado no arquivo e não mais poderá ser corrigido. Por último, vem o nome do equipamento onde será criado o arquivo; teclamos D para disco ou F para fita.

A próxima sub-rotina é montagem dos gabaritos. Ai temos uma facilidade, já que, após ter sido processada uma turma, o programa ainda guarda o gabarito anterior e pergunta se, para processar uma nova turma, devemos utilizá-lo. Caso negativo, teclamos S à pergunta sobre a utilização ou não de outro gabarito.

Para montarmos o gabarito, devemos teclar a letra correspondente à resposta

```

340 I=I+1
350 GOTO 310
360 REM
370 REM
380 REM
390 NQ=I-1
400 CLS
410 PRINT TAB(20)"A PROVA TEM ";NQ;"QUESTOES"
420 PRINT TAB(21)" CONFIRA-AS NOVAMENTE"
430 FOR L=1 TO 900:NEXT L:CLS
440 REM
450 REM
460 REM
470 Y=0
480 GOSUB 1990
490 CLS:PRINT 13-INT(LEN(NM$)/2),"GABARITO PARA A DISCIPLINA DE ";NM$;"
MONTADO";:FOR L=0 TO 900:NEXT L:G$="F"
500 REM
510 REM
520 REM
530 A=0
540 A=A+1
550 CLS
560 PRINT@64,"ENTRE NOME DO ALUNO";
570 INPUT M$(A,2):IF LEN(M$(A,2))=0 THEN 560
580 IF M$(A,2)="FIM" THEN GOTO 820
590 IF LEN(M$(A,2))>23 PRINT@896,"NUMERO DE CARACTERES DO NOME MAIOR D
O QUE 23, ENTRE NOVAMENTE":FOR L=0 TO 900:NEXT L:GOTO 550
600 CLS
610 PRINT@25-INT(LEN(M$(A,2))/2),"PROVA DO ALUNO ";M$(A,2);
630 FOR I=1 TO NQ
640 PRINT@245," "":PRINT@ 206,"ENTRE COM RESPOSTA PARA A QUESTAO ";I;
650 INPUT M$(I,4)
660 NEXT I
670 REM
680 REM
690 REM
700 REM
710 REM
720 H1$="RESPOSTAS DO ALUNO "":H2$=M$(A,2):X=4
730 Y=0:GOSUB 1990
740 REM
750 REM
760 REM
765 RC=0: FOR I=1 TO NQ: IF M$(I,4)<>M$(I,0) THEN 746 ELSE RC=RC+1:M$(I,
1)=STR$(VAL(M$(I,1))+1)
766 NEXT I
770 M$(A,3)=STR$(RC*(10/NQ))
780 GOTO 540
790 REM
800 REM
810 REM
820 A=A-1:GOSUB 2430
830 REM
840 REM
850 REM
860 GOSUB 1830
870 REM
880 REM
890 REM
900 REM
910 IF DV$="D" GOSUB 1420 ELSE GOSUB 1620
920 IF DV$="D" GOSUB 1510 ELSE GOSUB 1710
930 REM
940 REM
950 REM
960 REM
970 FOR I=1 TO NQ:M$(I,3)=STR$(VAL(M$(I,1))*100/A):NEXT I
980 H1$="INDICE DE APROVEITAMENTO DAS QUESTOES":H2$=" "":H3$="QUESTAO":H4
$=" "":Z=X-3:Y=1
990 GOSUB 1990
1000 REM
1010 REM
1020 REM
1030 GOTO 140
1040 REM
1050 REM
1055 REM
1060 REM
1065 REM
1070 REM
1080 CLS: PRINT @ 10, "LISTA NO (V)IDEO OU NA (I)MPRESSORA ? "":INPUT OS
:OS=LEFT$(OS,1)
1090 IF OS="V" OR OS="I" THEN 1100 ELSE 1080
1094 REM
1095 REM
1096 REM
1100 GOSUB 2500
1104 REM
1105 REM
1106 REM
1110 IF DV$="D" GOSUB 2600 ELSE GOSUB 2780
1115 REM
1120 GOSUB 2430
1125 REM

```


correta à questão indicada, seguido de ENTER. Quando todas as respostas tiverem sido entradas, teclamos FIM e ENTER. Feito isto, na primeira linha do vídeo será mostrado o número de questões da prova. Todo gabarito será, então, enviado para o vídeo em páginas com dez questões cada e poderão ser conferidas novamente. Para corrigir alguma resposta que porventura tenha sido introduzida erroneamente, basta pressionar qualquer tecla, exceto ENTER. Será pedido, na última linha do vídeo, que seja fornecido o número da questão a ter sua resposta corrigida. Após isto, entramos com a resposta correta. A página é enviada novamente para o vídeo com a modificação realizada. Para mudar de página, teclamos sempre ENTER.

Feita a montagem dos gabaritos, passamos para a correção das provas dos alunos. Devemos entrar, então, com os nomes dos alunos, os quais não deverão ter mais do que 23 caracteres. Caso não haja mais provas a serem corrigidas, ao invés do nome devemos entrar com FIM ENTER.

Após todas as respostas terem sido introduzidas, elas serão novamente mostradas no vídeo, em páginas de dez questões cada. Podemos, então, corrigir a resposta de qualquer questão que tenha sido introduzida erroneamente, utilizando o mesmo procedimento usado durante a montagem do gabarito. A prova será corrigida e a nota do aluno calculada pela seguinte fórmula:

$$\text{NOTA} = \text{RESP. CORRETAS} * (10 / \text{Nº QUEST.})$$

Vejamos agora a sub-rotina de classificação de turma por aluno ou por nota. Quando o último aluno for processado e teclarmos FIM ENTER, seremos consultados sobre a possibilidade de arquivar a turma, classificada ou não. Se optarmos pela classificação, devemos teclar A para o caso de ela ser por aluno, ou N, para o caso de ser por nota. O número de alunos da turma será mostrado na primeira linha do vídeo e no meio deste a palavra CLASSIFICANDO, que ficará piscando durante a execução.

Esta rotina é simples e, considerando que foi escrita em BASIC interpretado, levará um tempo razoável para ser completada. Foram realizados dois testes. No primeiro, com uma classificação de 52 alunos, o tempo requerido foi de 30 segundos; com uma turma de 211 alunos, o tempo necessário foi de 30 minutos.

Após a classificação, se realizada, a turma, os alunos e as notas, serão mos-

```

1130 IF O$="V" GOSUB 1830 ELSE GOSUB 2930
1140 CLS: PRINT@ 10, "OUTRA TURMA PARA LISTAR (S/N) ? "; INPUT R$: IF LEFT$(R$,1)="S" GOTO 1100 ELSE GOTO 140
1150 REM
1160 REM
1165 REM
1170 REM
1175 REM
1180 REM
1184 REM
1185 REM
1186 REM
1190 INPUT "ENTRADA EM (D)ISCO OU (F)ITA"; ID$: ID$=LEFT$(ID$,1)
1200 IF ID$="D" OR ID$="F" 1210 ELSE 1190
1210 INPUT "SAIDA EM (D)ISCO OU (F)ITA"; OD$: OD$=LEFT$(OD$,1)
1220 IF OD$="D" OR OD$="F" 1230 ELSE 1210
1230 IF ID$="D" DN$=" DO DISCO" ELSE DN$=" DA FITA"
1240 PRINT "ENTRE NOME DA TURMA A SER LIDA"; DN$; INPUT NT$
1250 IF OD$="D" DN$=" NO DISCO" ELSE DN$=" NA FITA"
1260 PRINT "ENTRE NOME DA TURMA A SER GRAVADA"; DN$; INPUT TN$
1270 INPUT "VOCE QUER SAIDA CLASSIFICADA (S/N) "; R$: IF LEFT$(R$,1)="S" THEN PRINT M1$; INPUT R1$
1280 IF ID$="D" INPUT "ENTRE NUMERO DO DISCO A SER LIDO"; DI$
1290 IF OD$="D" INPUT "ENTRE NUMERO DO DISCO A SER GRAVADO"; DO$
1294 REM
1295 REM
1296 REM
1300 IF ID$="D" THEN DN$=DI$: GOSUB 2650 ELSE GOSUB 2800
1320 REM
1330 REM
1330 REM
1340 REM
1350 GOSUB 2470
1360 REM
1370 REM
1370 REM
1380 REM
1385 NT$=TN$: IF OD$="D" THEN DN$=DO$: GOSUB 1470 ELSE GOSUB 1620
1390 IF OD$="D" GOSUB 1510 ELSE GOSUB 1710
1400 REM
1410 CLS: INPUT "OUTRA TRANSFERENCIA (S/N)"; R$: IF LEFT$(R$,1)="S" THEN GOTO 1190 ELSE GOTO 140
1414 REM
1415 REM
1416 REM
1420 REM
1430 REM
1440 REM
1450 REM
1460 CLS: INPUT "ENTRE NUMERO DO DISCO (0/1)"; DN$
1470 AR$="T"+NT$+" "+DN$
1480 OPEN "O",1,AR$
1490 PRINT #1,NT$," ",NM$
1500 RETURN
1510 REM
1520 REM
1530 REM
1540 REM
1550 CLS: PRINT@ 460, "GRAVANDO DADOS NO DISCO"
1560 FOR I=1 TO A
1570 PRINT #1,M$(IX(I),2);",",M$(IX(I),3)
1580 PRINT@ 32, USING "###"; I;
1590 NEXT I
1600 CLOSE #1
1610 RETURN
1620 REM
1630 REM
1640 REM
1650 REM
1660 CLS: PRINT "COLOQUE O GRAVADOR PARA GRAVAR TECLE (P)RONT0";
1670 INPUT R2$: IF LEFT$(R2$,1)("<P" GOTO 1660
1680 CT$=A$+NT$+A$: CM$=A$+NM$+A$: CA$=A$+M$(I,2)+A$
1690 PRINT #1,CT$,CM$,CA$,M$(I,3)
1700 RETURN
1710 REM
1720 REM
1730 REM
1740 REM
1750 CLS: PRINT@ 460, "GRAVANDO DADOS NA FITA"
1760 FOR I=1 TO A
1770 CA$=A$+M$(IX(I),2)+A$: CN$=M$(IX(I),3)
1780 PRINT #1,CT$,CM$,CA$,CN$
1790 PRINT@ 32, USING "###"; I;
1800 NEXT I
1810 PRINT #1,"FIM",0,0,0
1820 RETURN
1830 REM
1840 REM
1850 REM
1860 CLS: POKE 16916,3
1870 PRINT@ 14-INT(LEN(NT$)/2),"NOTAS TIRADAS PELOS ALUNOS DA TURMA "; NT$;
1880 PRINT@ 87-INT(LEN(NM$)/2),"NA DISCIPLINA DE "; NM$
1890 PRINT@ 143,"ALUNO"; @169,"NOTA"
1900 NP=INT(A/10): P=0

```

trados no vídeo, também por páginas, com dez alunos cada. Neste ponto, não podemos mais efetuar nenhuma alteração dos dados. Pressionamos ENTER para mudar de página.

Na sub-rotina da gravação da turma, se o arquivo for em disco, o programa perguntará que unidade (0 ou 1) deverá ser utilizada. Se o arquivo de saída for em fita, o programa solicitará que preparemos o gravador. Feito isto, teclamos P. Durante a gravação será mostrado, no meio da primeira linha do vídeo, o número do registro que está sendo gravado, sendo um registro por aluno.

Por fim, chegamos à última sub-rotina para a criação do arquivo, que é o índice de aproveitamento das questões. Após a gravação da turma, será mostrado no vídeo esse índice, por páginas, com dez questões cada. Para mudarmos de página, teclamos sempre ENTER. Este índice é importante para que o professor saiba quais as questões que apresentaram maior nível de dificuldade ou facilidade para a turma, de modo que ele possa rever a matéria concernente àquelas. Com isto termina o processamento da turma e, se mais nenhuma correção for requisitada, o programa volta ao seu início.

PARA (L)ISTAR

Após a criação do arquivo, podemos listá-lo. Para isto, basta tecarmos L. A saída poderá ser realizada no vídeo ou na impressora. Serão pedidos, então, os mesmos dados solicitados inicialmente pela rotina de criação (correção de provas) de arquivo, ou sejam: nome da turma, nome da disciplina e o equipamento onde se encontra o arquivo (disco ou fita).

Se o arquivo estiver em fita, devemos colocar o gravador pronto para leitura; se estiver em disco, devemos fornecer a unidade que deve ser utilizada (0 ou 1). Durante a leitura, o número do registro será mostrado no meio da primeira linha do vídeo. Após a leitura do arquivo, podemos escolher entre a saída classificada (por aluno ou nota) ou não.

A turma é então mostrada no vídeo, em páginas, cada uma com dez alunos. Se a saída for na impressora, os dados serão enviados a ela. Ao fim desse trabalho, se mais nenhuma listagem for requerida, o programa retorna ao seu início, solicitando outra tarefa a ser realizada.

```

1910 IF P*10+10>A THEN LL=A ELSE LL=P*10+10
1920 FOR I=P*10+1 TO LL
1930 PRINT M$(IX(I),2);TAB(40) USING "###.##";VAL(M$(IX(I),3))
1940 NEXT I
1950 R$=INKEY$:IF R$="" GOTO 1950
1960 IF ASC(R$)=13 AND LL<A THEN P=P+1:CLS:GOTO1910
1970 POKE 16916,0
1980 RETURN
1990 REM
2000 REM      SUB-ROTINA DE DISPLAY POR PAGINAS
2010 REM
2020 CLS:POKE 16916,3
2030 PRINT$ 32-INT((LEN(H1$)+LEN(H2$))/2),H1$,H2$
2040 PRINT$ 87,H3$
2050 PRINT$ 97,H4$
2060 PRINT
2070 NP=INT(NQ/10):P=0
2080 IF P*10+10>NQ THEN LL=NQ ELSE LL=P*10+10
2090 FOR I=P*10+1 TO LL
2100 PRINT TAB(25) USING"###.##";I;:IF VAL(M$(I,X))=0 THEN PRINT TAB(37) M$
(I,X) ELSE PRINT TAB(37) USING "###.##";VAL(M$(I,X))
2110 NEXT I
2120 R$=INKEY$:IF R$="" GOTO2120
2130 IF ASC(R$)=13 AND LL<NQ THEN P=P+1:CLS:GOTO 2080
2140 IF ASC(R$)=13 GOTO 2210
2145 IF Y=1 GOTO 2120
2150 PRINT$917,"      ":PRINT$896,"ENTRE NUMERO QUESTAO";
2160 INPUT R$:I=VAL(R$)
2170 IF VAL(R$)<LL AND VAL(R$)>P*10+1 THEN 2190
2180 PRINT$ 896,"NUMERO DA QUESTAO FORA DA PAGINA, ENTRE NOVAMENTE":FOR
L=0 TO 900:NEXT L:PRINT$ 896,STRING$(64," ");:FOR L=0 TO 900:NEXT L:GOTO
2150
2190 PRINT$917,"      ":PRINT$896,"ENTRE RESPOSTA CORRETA";
2200 INPUT M$(I,X):CLS:GOTO 2080
2210 POKE 16916,0
2220 RETURN
2230 REM
2240 REM      SUB-ROTINA DE CLASSIFICACAO
2250 REM
2270 R$=LEFT$(R1$,1):IF R$="A" THEN J=2 ELSE IF R$="N" THEN J=3
2280 CLS:PRINT$ 20,"O NUMERO DE ALUNOS E ";A
2290 PRINT$ 468,"C L A S S I F I C A N D O"
2300 FOR IC=1 TO A-1
2310 PRINT$ 468,"
2320 FOR IM=IC+1 TO A
2330 PRINT$ 468,"C L A S S I F I C A N D O"
2340 IF VAL(M$(IX(IC),J)) AND VAL(M$(IX(IM),J)) <>0 GOTO2360
2350 IF M$(IX(IC),J)<M$(IX(IM),J) GOTO 2400 ELSE 2370
2360 IF VAL(M$(IX(IC),J))=VAL(M$(IX(IM),J)) GOTO 2400
2370 IX(0)=IX(IM)
2380 IX(IM)=IX(IC)
2390 IX(IC)=IX(0)
2400 NEXT IM
2410 NEXT IC
2420 RETURN
2430 REM
2440 REM      SUB-ROTINA DE PREPARACAO PARA CLASSIFICACAO
2450 REM
2460 CLS:PRINT$ 10,"VOCE QUER SAIDA CLASSIFICADA (S/N) ":INPUT R$
2465 IF LEFT$(R$,1)="N" THEN 2470 ELSE PRINT$ 10,M1$;:INPUT R1$
2470 CLS:FOR I=0 TO A:IX(I)=I:NEXT I
2480 IF LEFT$(R$,1)="S" AND A>1 THEN GOSUB 2230 ELSE 2490
2490 RETURN
2500 REM
2510 REM      SUB-ROTINA DE ENTRADA DE DADOS
2520 REM
2530 INPUT "ENTRE NOME DA TURMA ";NT$
2540 IF LEN(NT$)>11 THEN 2530
2550 A=0
2560 INPUT "ENTRE NOME DA DISCIPLINA ";NM$:IF LEN(NM$)>24 THEN PRINT "O
NOME DA DISCIPLINA DEVE TER ATÉ 24 CARACTERES":GOTO 2560
2570 INPUT "ARQUIVO EM (D)ISCO OU (F)ITA ";DV$:DV$=LEFT$(DV$,1)
2580 IF DV$="D" OR DV$="F" THEN 2590 ELSE 2570
2590 RETURN
2600 REM
2610 REM      SUB-ROTINA DE LEITURA DE ARQUIVO DE DISCO
2620 REM
2630 INPUT "ENTRE NUMERO DO DISCO (0/1) ";DN$
2640 IF DN$="0" OR DN$="1" THEN 2650 ELSE 2630
2650 CLS:PRINT$ 460,"L E N D O D A D O S D O D I S C O"
2660 AR$="I"+NT$+" "+DN$
2670 OPEN "I",1,AR$
2680 I=0
2690 IF EOF(1) GOTO 2750
2700 INPUT #1,M$(I,2),M$(I,3)
2710 PRINT$32,USING"###.##";I;
2720 IF I=0 THEN NM$=M$(I,3)
2730 I=I+1
2740 GOTO 2690
2750 A=I-1
2760 CLOSE
2770 RETURN
2780 REM

```


PARA (T) RANSFERIR

Esta rotina possibilita transferir arquivos de um equipamento para outro, do disco para fita e vice-versa, para o mesmo equipamento, de um disco para outro ou de uma fita para outra.

Para efetuarmos a transferência, são pedidos os seguintes dados:

- equipamento onde está o arquivo a ser transferido, disco ou fita;
- o equipamento para onde será transferido o arquivo, disco ou fita;
- o nome da turma a ser transferida;
- o nome da turma a ser gravado no novo arquivo;
- se a saída deve ser classificada (por aluno ou nota) ou não;
- se o equipamento de entrada e/ou saída for disco, qual a unidade que será usada para leitura e/ou gravação do arquivo.

O motivo da existência da quarta pergunta (o nome da turma a ser gravado no arquivo de saída) é possibilitar a mu-

dança do nome, se quisermos corrigi-lo ou renomeá-lo. Caso a classificação não seja pedida, o arquivo de saída conterà os dados na mesma sequência em que foram lidos. Para eliminar um arquivo do disco, devemos utilizar o comando KILL do DOS. Quando nenhuma outra transferência for requerida, o programa volta à pergunta inicial.

```

2790 REM          SUB-ROTINA DE LEITURA DE ARQUIVO DE FITA
2800 REM
2810 PRINT "COLOQUE O GRAVADOR PARA LEITURA TECLE (P)RONT0";
2820 INPUT R2$;
2830 CLS:PRINT@460,"L E N D O  D A D O S  D A  F I T A"
2840 I=0
2850 INPUT #-1,CT$,CM$,M$(I,2),M$(I,3)
2860 IF CT$="FIM" AND I=0 GOTO 2850 ELSE IF CT$="FIM" THEN 2920
2870 IF I=0 PRINT@960,STRING$(63," ");:PRINT@ 981-INT((LEN(CT$)+LEN(CM$)
)/2),"LEND0 ",CT$," DISCIPLINA DE ",CM$;
2880 IF I=0 AND T$="T" AND CT$=NT$ THEN NM$=CM$: GOTO 2900
2890 IF CT$=NT$ AND CM$=NM$ THEN 2900 ELSE 2850
2900 PRINT@ 32,USING "###";I;
2910 I=I+1: GOTO 2850
2920 A=I-1:RETURN
2930 REM
2940 REM          SUB-ROTINA DE SAIDA EM IMPRESSORA
2950 REM
2960 LPRINT:LPRINT TAB(10)"NOTAS TIRADAS PELOS ALUNOS DA TURMA ",NT$
2970 LPRINT TAB(15) "NA DISCIPLINA DE ";NM$
2980 LPRINT TAB(20)"ALUNO",TAB(41)"NOTA";
2990 LPRINT:LPRINT:
3000 FOR I=1 TO A
3010 LPRINT M$(IX(I),2);TAB(40) USING"###.##";VAL(M$(IX(I),3))
3020 NEXT I
3030 CLS:PRINT@ 468,"F I M  D A  I M P R E S S A 0"
3040 FOR L=D TO 900:NEXT L
3050 RETURN

```

Lawrence Falconer King é programador há cinco anos e meio, trabalhando na área de transmissão de dados internacionais, mais especificamente no SICRAM (Sistema Computarizado de Retransmissão Automática de Mensagens), tendo ministrado cursos de operação, software e linguagem Assembler relativos a esse sistema, na EMBRATEL.

← KSR BI-DIRECIONAL →

O único terminal teleimpressor fabricado no Brasil.

A ISA, Indústria de Impressoras S/A, acaba de lançar o primeiro teleimpressor genuinamente brasileiro. Sua alta tecnologia eletrônica, prova na prática sua capacidade de trabalho. Imprime por matriz de pontos 9x7, permitindo até 8 cópias impressas, com uma velocidade de 100 C.P.S. Equipado com memória standard de 2 K caracteres, utilizando formulário contínuo, folhas soltas ou bobina de papel. De 64 até 132 caracteres por linha, com interface serial RS 232 elo de corrente e paralelo. Sua velocidade de comunicação é de 50 a 9.600 B.P.S. 8 diferentes tipos de impressão.



INDÚSTRIA DE IMPRESSORAS S.A.

Escritório:
R. Prof. Jose Marques da Cruz, 234
Fabrica:
Rua Centro Africana, 74
Tels.: (011) 240.2442 - 543.4939
Telex: (011) 369261111 BR
São Paulo - S.P.



Maio-Tum

Conheça mais
detalhes do
Teleimpressor
KSR na

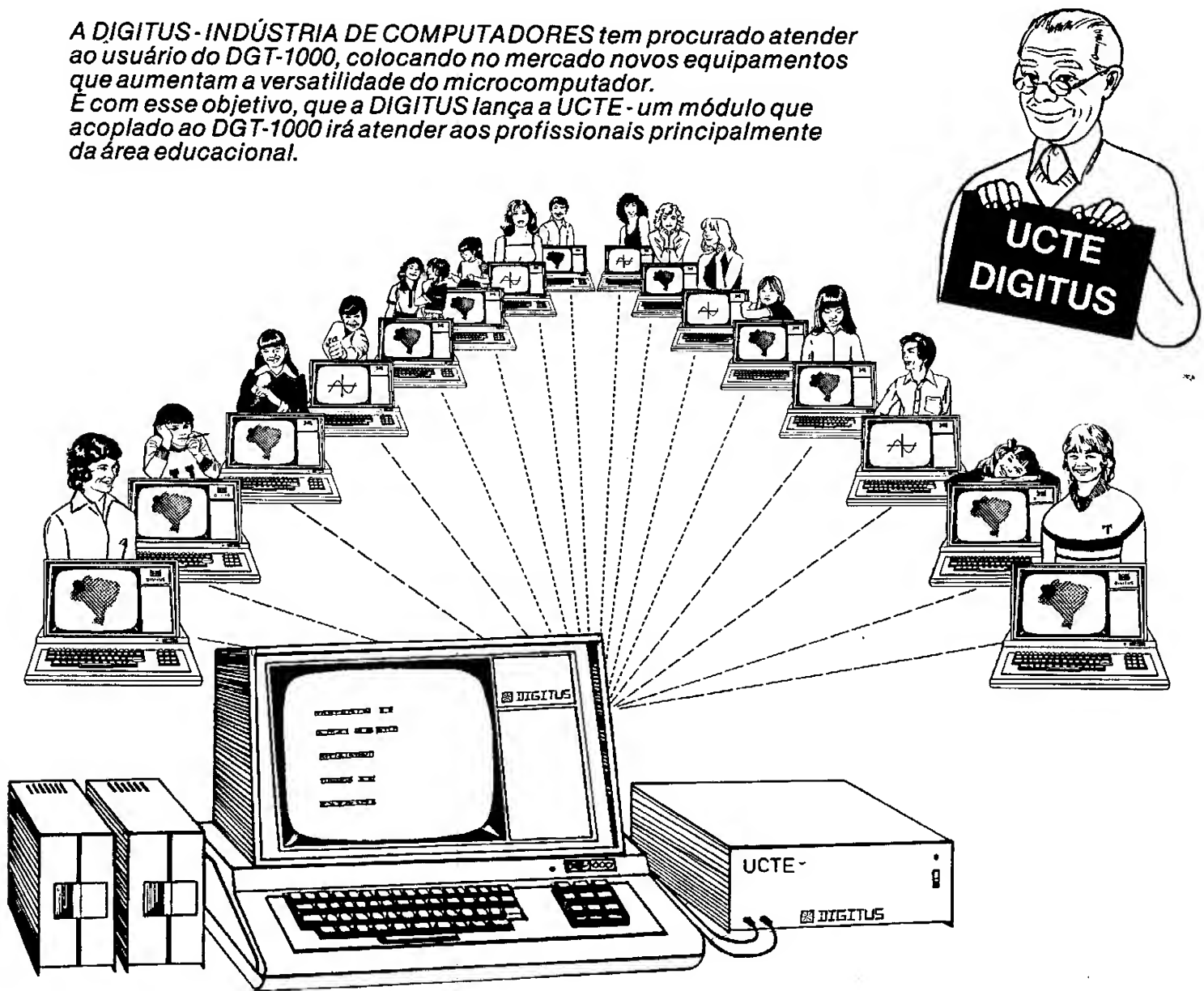
DECODATA
SISTEMA DE PROCESSAMENTO DE DADOS E TELECOMUNICAÇÕES

MATRIZ: Rio de Janeiro - R. República do Líbano, 61 - 12º and.
Tel.: (021) 221-4677 - Telex (021) 30187
FILIAIS: S. Paulo - R. Frei Caneca, 1119 Tel.: (011) 284-8311 - Telex (011) 22191
Brasília - SCS 02 - Bloco C nº 41 - SL 01 - Tel.: (061) 225-1588 - Telex (061) 1750
Porto Alegre - Rua Santa Terezinha, 300 - Tel.: (051) 32-3564 - Telex (051) 2144
Goiânia - Belo Horizonte - Salvador

ESCOLA DO FUTURO. HOJE!

A DIGITUS-INDÚSTRIA DE COMPUTADORES tem procurado atender ao usuário do DGT-1000, colocando no mercado novos equipamentos que aumentam a versatilidade do microcomputador.

É com esse objetivo, que a DIGITUS lança a UCTE - um módulo que acoplado ao DGT-1000 irá atender aos profissionais principalmente da área educacional.



UCTE - Unidade Controladora de Terminais Educacionais.

DIGITUS-
Rua Gávea, 150 Belo Horizonte
tel: (031) 332.8300 tx: 3352
Rio de Janeiro — RJ.
tel: (021) 257-2960

Esta unidade permite a comunicação entre um DGT-1000 central com até 16 (dezesesseis) DGT-1000 secundários através da porta de cassete.

Os dezesseis terminais poderão ser computadores na configuração mínima, pois sendo a comunicação através da porta de cassete não é necessário qualquer modificação nos terminais.

O computador central, comandado pelo instrutor, deverá ter uma configuração mínima de 48KB de memória e um Disk-Drive.

O sistema permite ao instrutor carregar programas nas estações individuais, podendo aplicar provas e ter as respostas automaticamente gravadas no disquete, para posterior avaliação.

Nas estações individuais, os estudantes poderão gravar e carregar programas diretamente no computador do instrutor e poderão também usar o seu computador independentemente.

Suas aplicações são de grande valia para centros educacionais, cursos monitorizados, escolas técnicas, etc.

A UCTE facilitará muito o aprendizado em cursos de programação e operação de computadores, além de permitir o ensino em outras áreas.

A UCTE dispensa o uso incômodo do cassete e o seu custo é menor do que comprar os gravadores para os terminais.

OBS.: A UCTE pode ser utilizada com o DGT-100. Consulte seu revendedor.

Conheça dois novos endereços de memória do Apple: um deles é capaz de produzir diversos sons; o outro, controla o entra-e-sai de informações das fitas cassete



Efeitos sonoros e clicks de gravação

Rudolf Horner Junior

Existe na memória de qualquer microcomputador uma série de endereços que são utilizados pelo próprio equipamento para registrar determinados valores lógicos do sistema. Manipulando-os diretamente, várias operações tornam-se excepcionalmente mais rápidas e fáceis de serem obtidas, e em algumas ocasiões podemos até realizar proezas impossíveis de serem alcançadas com a utilização única empregada em um determinado equipamento.

Qualquer bom programa escrito em BASIC — e que tenha um alto grau de eficiência — normalmente usa muitos endereços e rotinas em linguagem de máquina (comandos POKE, PEEK, CALL e URS). Muitos programadores, entretanto, refutam estas grandes vantagens por trazerem com elas um efeito colateral, quase sempre, bastante grave. Sistemas de software, que embora escritos em linguagem de alto nível usem endereços específicos de memória e rotinas montadas em Assembler na RAM ou já disponíveis na ROM de cada máquina, acabam não sendo portáteis de um equipamento para outro, ou seja, os programas, apesar de escritos em uma mesma linguagem, não podem ser rodados em equipamentos de linhas não compatíveis.

Para ilustrar vamos citar um exemplo. É bastante fácil desenvolver um programa em BASIC para uma máquina da linha Apple que crie um gráfico qualquer. O programa irá se tornar mais eficiente à medida em que utilizarmos algumas características específicas do equipamento. No entanto, ao mesmo tempo que o tornamos mais eficiente, dificultamos a sua execução

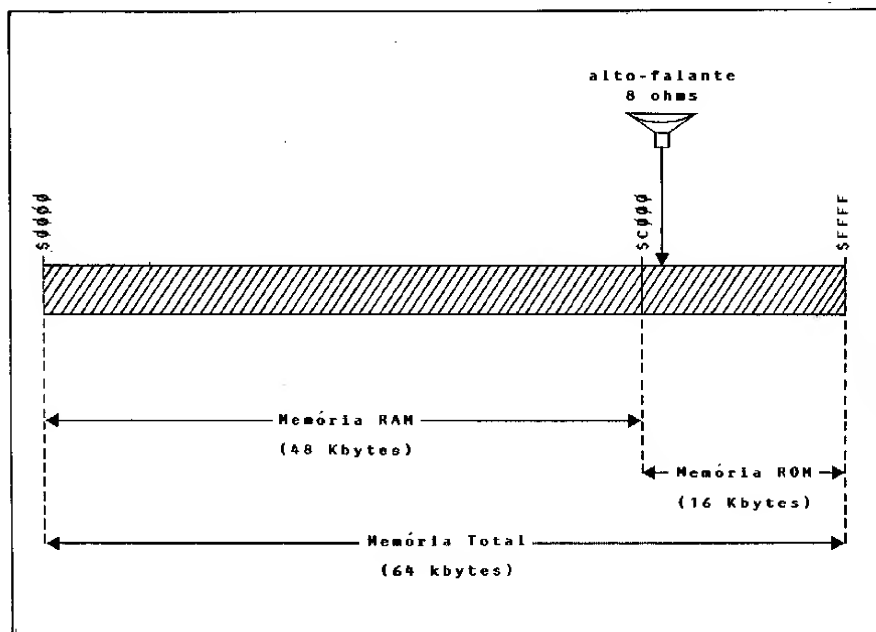


Figura 1

em um computador da linha TRS-80 ou qualquer outro cujas características não sejam idênticas às do Apple. Justamente para tentar evitar esta *Torre de Babel*, os sistemas criados para funcionar em diversos tipos de equipamento acabam perdendo eficiência para que a portabilidade não fique prejudicada.

Mas ainda bem que nossos objetivos nesta seção não são estes. Portanto, vamos falar sobre mais alguns endereços de memória que foram idealizados para que pudessem realizar algumas tarefas especiais.

O ENDEREÇO DO SOM

Como já sabemos, existem no Apple certos endereços de memória que atuam simplesmente como comutadores. São endereços que, quando referenciados em uma instrução de leitura ou escrita, fazem com que algum dispositivo do equipamento passe de um estado a outro. Creio que o exemplo mais fácil deste tipo de comutadores é o que está associado às páginas de alta resolução gráfica. Tratam-se de dois endereços de memória cuja simples referência em linguagem de alto ou baixo nível leva a tela do micro a exibir a página um ou a página dois de alta resolução de gráficos.

Desta vez, porém, falaremos sobre um outro endereço de memória que tem um funcionamento similar a estes dois. É mais um endereço constante no grupo das 4096 localizações de memória especialmente destinadas às funções de entrada e saída. Estamos falando do endereço \$C030 (em decimal, equivalente a 49200 ou -16336), o qual está diretamente vinculado a um pequeno alto-falante de oito ohms que vem dentro da caixa das máquinas nacionais compatíveis com o Apple II.

Veja na figura 1 o esquema que define a posição deste alto-falante. A localização \$C030 é um exemplo típico de um endereço comutador. A simples referência a esta localização faz com que o cone de papel do alto-falante alterne suas posições entre *para dentro* e *para fora*. A cada referência ao endereço, haverá uma mudança do estado do cone entre estas duas possibilidades, e cada vez que ocorre uma mudança de estado o alto-falante emite um pequeno *click*.

Os manuais que acompanham o Apple e similares explicam como obter sons referenciando o alto-falante pelo comando BASIC que todos, provavelmente, já usaram: PEEK (-16336) ou PEEK (49200). Qualquer um deles fará com que o alto-fa-

lante produza um som praticamente imperceptível. Já se colocarmos um destes comandos dentro de um *loop* que o execute diversas vezes, o resultado poderá ser um som cacofônico bastante fácil de ser percebido.

Contudo, torna-se difícil procurar emitir notas musicais com frequências precisas usando linguagem BASIC interpretada. É que o interpretador consome um tempo muito grande para executar o programa e nossas frequências acabam sendo comprometidas. Quando queremos utilizar efeitos sonoros mais interessantes em nossos programas, somos, via de regra, obrigados a trabalhar com rotinas em linguagem de máquina cujo tempo de execução é muito menor que o de programas em BASIC.

Para criarmos uma rotina desta natureza — e que satisfaça plenamente nossos objetivos — devemos ter sempre em mente o tempo necessário para a execução de cada comando existente na rotina; isto para que possamos fazer um número correto de referências ao alto-falante por unidade de tempo e produzir sons das mais variadas frequências.

É claro que devemos agir também com bom senso. Se você criar uma rotina que simplesmente referencie o endereço \$C030 e na instrução seguinte volte ao início, completando um *loop*, evidentemente você não poderá ouvir nenhum resultado. Não perca de vista o fato de que a frequência do relógio do equipamento é de 1.023 MHz e que, mesmo que o seu alto-falante conseguisse reagir e trocar de estados nesta velocidade, seu ouvido humano, por mais apurado que seja, não conseguiria ouvir sons com frequência superior a vinte mil Hz. Se a rotina funcionasse, ela só poderia produzir ultra-sons.

Vamos ver agora duas rotinas em linguagem de máquina que foram criadas para uso em programas de jogos e que tiveram de ser elaboradas dentro deste espírito. Veja inicialmente a lista em Assembler da figura 2. Temos uma rotina que servirá

AGORA ESTÁ MAIS FÁCIL ASSINAR MICRO SISTEMAS

Para sua maior comodidade, a ATI Editora Ltda. coloca à sua disposição os seguintes endereços de seus representantes autorizados:

RIO DE JANEIRO

ATI Editora Ltda.
Av. Presidente Wilson, 165 — GR. 1210
CEP 20030 — Tels.: (021) 262-5259
e 262-5208

SÃO PAULO

Embass Representações Ltda.
Rua Cel. Xavier de Toledo, 210 — Cj. 23
CEP 01048 — Tel.: (011) 34-8391

BELO HORIZONTE

Profissional Com. Rep. Editoriais Ltda.
Rua Guajajaras, 410 — Cj. 305
CEP 30000 — Tel.: (031) 222-8679

PORTO ALEGRE

Aurora Assessoria Empresarial Ltda.
Rua Uruguai, 35 — sala 622
CEP 90000 — Tel.: (0512) 26-0839

RECIFE

Monte São Distr. Nordeste Ltda.
Rua Almeida Cunha, 65
CEP 50000 — Tel.: (081) 222-1699

GOIÂNIA

Tiago Motta Araujo
Rua 6, nº 310 — CEP 74000

EFEITOS SONOROS E CLICKS DE GRAVAÇÃO

```

0300- 20 09 03 JSR $0309
0303- 20 09 03 JSR $0309
0306- 20 09 03 JSR $0309
0309- A9 00 LDA #$00
030B- 85 FF STA $FF
030D- A9 FF LDA #$FF
030F- 85 FE STA $FE
0311- A9 00 LDA #$00
0313- 30 C0 BMI $02D5
0315- EE 30 C0 INC $C030
0318- CE 30 CD DEC $CD30
031B- A6 FF LDX $FF
031D- CA DEX
031E- D0 FD BNE $031D
0320- C6 FE DEC $FE
0322- F0 05 BEQ $0329
0324- E6 FF INC $FF
0326- 4C 11 03 JMP $0311
0329- 60 RTS
032A- 00 BRK
    
```

Figura 2

para produzir o disparo de uma arma futurista, um revólver de raio laser. Observe, na listagem, as referências ao endereço \$C030. Veja também porque a rotina, quando executada, dará a impressão de um som com diversos pulsos. As três primeiras instruções contêm uma chamada à própria sub-rotina de criação do efeito sonoro. Por este motivo, ele será executado várias vezes.

Veja, na figura 3, a listagem de um programa em BASIC que construirá a rotina para você. Carregue-o na memória da forma como aparece listado, execute-o com um comando RUN e logo após use o comando CALL 768 para ouvir o disparo do revólver.

Experimentada esta rotina, vamos tentar uma um pouco mais complicada. Vamos carregar a nova rotina na mesma posição que a anterior: a terceira página da memória RAM do Apple. A nova rotina está listada na figura 4. Como você pode

```

100 DATA 32,9,3,32,9,3,32,9,3,16
    9,0,133,255,169,255,133,254,
    169,0,48,192,238,48,192,206,
    48,205,166,255,202,208,253,1
    98,254,240,5,230,255,76,17,3
    ,96
110 FOR A = 768 TO 809: READ B: POKE
    A,B: NEXT : END
    
```

Figura 3

ver, ela é bem mais cheia de detalhes que a anterior. O objetivo desta vez é obter o efeito sonoro de um helicóptero voando.

Se preferir o BASIC, use o programa da figura 5: ele carregará a rotina para você. Digite-o exatamente como está listado, execute-o com o comando RUN e, em seguida, execute a rotina em linguagem de máquina que foi carregada na memória. Comande CALL 768 e ouça, por alguns instantes, o som da hélice de um helicóptero golpeando o ar.

Aí está. Não existe limite para os efeitos que podem ser obtidos com o uso do endereço de memória vinculado ao alto-falante. Basta apenas um pouco de cuidado e imaginação.

REGISTRO DE INFORMAÇÕES EM FITA

Vamos discutir agora o processo utilizado pelo Apple para registrar informações em uma fita magnética. Este processo tem a ver com o nosso assunto pois trata-se do uso de uma localização de memória com funcionamento similar ao que acabamos de descrever.

Todos sabemos que na parte traseira de nosso micro temos dois conectores denominados *IN* e *OUT*, cuja função é, respectivamente, receber e enviar informações para um equipamento de gravação em fita magnética, um gravador cassete por exemplo. Na realidade, ao conector *OUT* é associado um endereço de memória da mesma forma como é associada uma localização ao alto-falante. O endereço associado ao conector *OUT* corresponde à localização \$C020 (em decimal, 49184 ou -16352).

Fazer referência a esta posição faz com que a voltagem do conector *OUT* mude de zero para 250 mv. Se existir um equi-

```

0300- A0 23 LDY #$23
0302- 20 08 03 JSR $0308
0305- 88 DEY
0306- D0 FA BNE $0302
0308- A9 01 LDA #$01
030A- 85 FF STA $FF
030C- A9 45 LDA #$45
030E- 85 FE STA $FE
0310- A9 00 LDA #$00
0312- 30 30 C0 STA $C030
0315- EE 30 C0 INC $C030
0318- A6 FF LDX $FF
031A- CA DEX
031B- D0 FD BNE $031A
031D- C6 FE DEC $FE
031F- F0 05 BEQ $0326
0321- E6 FF INC $FF
0323- 4C 10 03 JMP $0310
0326- A9 12 LDA #$12
0328- 85 08 STA $08
032A- A9 20 LDA #$20
032C- 85 06 STA $06
032E- A9 03 LDA #$03
0330- 85 07 STA $07
0332- 30 30 C0 STA $C030
0335- EE 30 00 INC $0030
0338- CE 30 C0 DEC $C030
033B- C6 06 DEC $06
033D- D0 02 BNE $0341
033F- C6 07 DEC $07
0341- D0 F8 BNE $033B
0343- C6 08 DEC $08
0345- D0 E3 BNE $032A
0347- 60 RTS
0348- 00 BRK
0349- 00 BRK
034A- 00 BRK
034B- 00 BRK
034C- 00 BRK
034D- 00 BRK
    
```

Figura 4

```

100 DATA 160,35,32,8,3,136,208,2
    50,169,1,133,255,169,69,133,
    254,169,0,141,48,192,238,48,
    192,166,255,202,208,253,198,
    254,240,5
110 DATA 230,255,76,16,3,169,18,
    133,8,169,32,133,6,169,3,133
    ,7,141,48,192,238,48,0,206,4
    8,192,198,6,208,2,198,7,208,
    248,198,8,208,227,96
120 FOR A = 768 TO 839: READ B: POKE
    A,B: NEXT : END

```

Figura 5

pamento de gravação, será registrado em fita um pequeno *click*. Da mesma forma que na produção de música, ao referenciarmos a localização \$C020 contínua e repetidamente, poderemos produzir, na fita, um tom de gravação. Pela variação de duração e amplitude deste tom, a informação poderá ser codificada.

Existem, no programa monitor da ROM, rotinas para codificar informações em fita cassete que são usadas para executar os comandos do monitor **READ** e **WRITE**. As mesmas rotinas são utilizadas para executar os comandos **LOAD** e **SAVE** do BASIC Applesoft.

Para o procedimento de leitura de informações, temos o processo inverso. Uma rotina em linguagem de máquina verifica o sinal existente no conector *IN* e, através da inspeção da localização de memória \$C060 (em decimal equivale a 49248

ou -16288), pode-se ouvir os tons que foram originalmente gravados na fita. Da interpretação destes tons, o computador pode recuperar as informações que haviam sido registradas anteriormente.

Como vimos, devemos usar, portanto, os endereços \$C020 para registrar as informações e \$C060 para recuperá-las. É importante, porém, observar um detalhe: para que se produza um *click* no conector *OUT*, basta que se faça uma referência ao endereço \$C020. Esta referência poderia ser de duas espécies: uma de leitura (em Assembler, **LDA**; em BASIC, **PEEK**) e outra de escrita (em Assembler, **STA**; em BASIC, **POKE**).

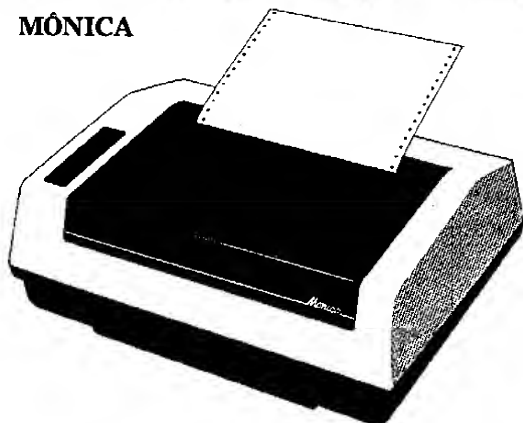
Uma referência de escrita não poderia ser usada pelo processador 6502 (devido ao sistema utilizado) para gravar dados numa localização de memória: primeiro ele lê o conteúdo desta localização para depois gravar nela o valor desejado. Isto significa uma dupla referência e, conseqüentemente, dois *clicks* ao invés de apenas um como seria correto.

É claro que com estes endereços poderíamos criar as nossas próprias rotinas para registrar e recuperar informações de um gravador de fita. No entanto, estas não poderiam estar em linguagem BASIC, uma vez que o interpretador não seria suficientemente rápido para ler e interpretar as variações dos estados das localizações de memória associadas aos conectores *IN* e *OUT*. As rotinas teriam que estar escritas em linguagem de máquina.

Rudolf Horner Junior cursa Ciência da Computação na Unicamp e é sócio da Potencial Software, firma que produz programas especiais para microcomputadores em Campinas, SP.

UM PROGRAMA COM BOA COMPANHIA

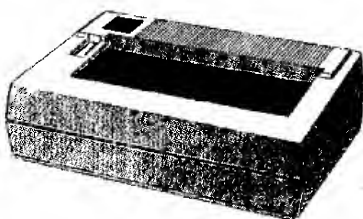
MÔNICA



80 posições de impressão (normal), bidirecional, velocidade de 100cps, densidade horizontal 5 a 16,7 cpp, form. em até 5 vias. Interface paralela. Módulo opcional de alta resolução gráfica "bit image" (endereçamento das agulhas) e qualidade de impressão carta.

A IMPRESSORA COMPLETA E DE MENOR CUSTO DO MERCADO.

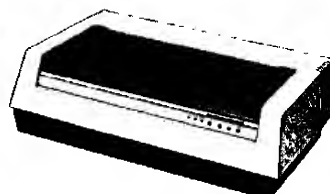
EMÍLIA



132 posições de impressão (normal), bidirecional, velocidade de 100cps, densidade horizontal 5 a 20 cpp, form. em até 5 vias. Capacidade gráfica de 72 x 75 pts por polegada. Interface paralela ou serial.

PARA QUEM NÃO PODE ABRIR MÃO DE FORMULÁRIOS DE 132 POSIÇÕES.

ALICE



132 posições de impressão (normal), bidirecional, velocidade de 200 cps, densidade horizontal 5 ou 10 cpp, form. em até 7 vias. Interface paralela ou serial.

PARA MICROS OU MINICOMPUTADORES, O COMPROMISSO DE VELOCIDADE E ROBUSTEZ.

repro

VENDA-LEASING-ALUGUEL

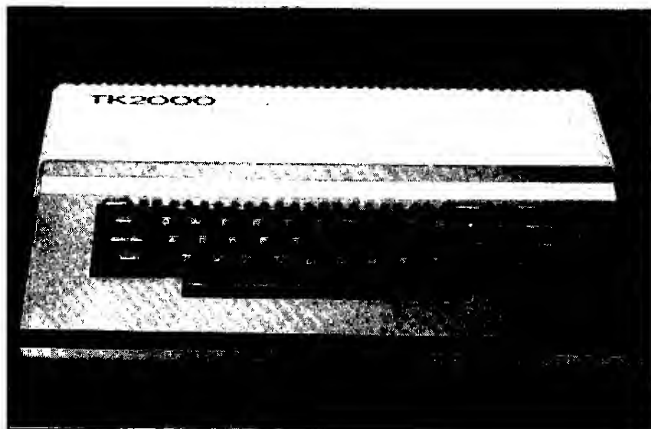
PRONTA ENTREGA DE TODOS OS MODELOS

ELEBRA
INFORMÁTICA

compumicro
INFORMÁTICA EMPRESARIAL LTDA.

Rua Sete de Setembro, 99/11.º andar
CEP 20050 Tel.: PBX (021) 224-7007

TK2000 no mercado



TK 2000

Apresentado ao público durante a Feira de Informática do ano passado, se encontra agora disponível no mercado o TK 2000, computador colorido da Microdigital. O 2000 não é totalmente compatível com o Apple, mas poderá rodar programas gerados em BASIC para essa linha de computadores desde que sejam feitas algumas mudanças nos endereços da ROM. O equipamento possui dois formatos de gravação em fita cassete: o seu próprio formato e outro compatível com o da linha Apple.

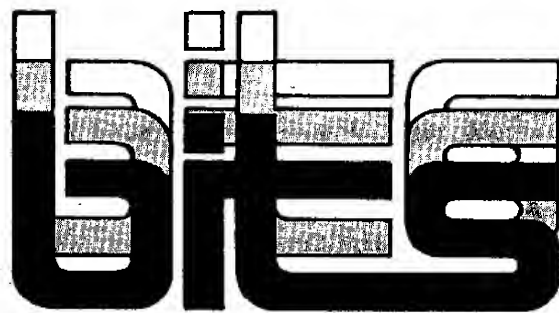
A UCP do novo micro trabalha com microprocessador 6502, de 8 bits. O TK 2000 tem 16 Kbytes de memória ROM e 64 Kbytes de memória RAM, disponíveis para o usuário, seu teclado é profissional, com 54 teclas, e o novo equipamento da Microdigital pode ser ligado diretamente a uma televisão colorida, sem que seja necessária qualquer alteração na TV. O 2000 possui saída para sinal de vídeo, podendo ser ligado a um monitor profissional, e também saída de um canal de som pela

TV. O display do vídeo é apresentado em quatro modos. No modo texto, com formato de 40 caracteres X 24 linhas; modo gráfico com formato de 40x48; modo gráfico de alta resolução com formato de 280x192 pontos e display texto de alta resolução. Em sua configuração básica, o TK 2000 vem com saída RF, saída para monitor de vídeo preto e branco ou a cores, saída para gravador cassete e interface paralela tipo Centronics para impressora. Na versão básica, possui monitor Disassembler e mini-assembler em ROM. O novo equipamento da Microdigital pode ser encontrado em várias lojas de revenda de microcomputadores e seu preço é de Cr\$ 849.850,00 (preço de abril). Quanto ao software, para o 2000, já se encontram disponíveis os seguintes programas: Controle de Estoque; Cadastro de Clientes e Mala Direta; Contas a Pagar; Contas a Receber; Controle Bancário; Multiplan, e os jogos, multi-invader; sabotagem; pânico; auto-estrada; corrida; papatudo e ataque, todos da Micro-soft.

Pacote topográfico da Codimex

Seguindo sua política de apoio ao usuário profissional, a Codimex está oferecendo, a nível nacional, um pacote destinado a aplicações na área de Topografia, o qual inclui um micro Codimex-6809 com 32 Kb e uma unidade de disco, uma impressora Ecodata bidirecional com 100 CPS e progra-

mas para Cálculo de Poligonal, Cálculo de Área e outros específicos que *descobrem* erros eventuais de cálculo de ângulo e medida cometidos no campo. Para maiores informações, o endereço da Codimex é Av. Wenceslau Escobar, 1549, Vila Assunção, Porto Alegre, RS.



A família I 7000

O novo integrante da família I 7000, da Itaútec, o Júnior, apresentado ao público durante a última feira de Informática, teve seu lançamento comercial no final de março. O novo microcomputador da Itaútec foi desenvolvido em duas versões: I 7000 Júnior e I 7000 Júnior E. A diferença básica entre os dois modelos está no teclado. Além do teclado tipo máquina de escrever com caracteres em português, do modelo mais simples, o Júnior E possui também um teclado numérico separado e pode trabalhar com disquetes de 5" ou 8". Juntamente com o lançamento do Júnior, a Itaútec apresentou um conjunto de serviços tais como assistência técnica; centro educacional para cursos e treinamento; centro de atendimento ao usuário, para atender as dúvidas de clientes por telefone ou telex e um catálogo com 161 programas aplicativos desenvol-

vidos por 44 software-houses credenciadas pela Itaútec. Os três equipamentos da Itaútec são totalmente compatíveis entre si, e a migração do Júnior para o Júnior E e para o I 7000 se dá através de um kit de expansão. O I 7000 Júnior, em seu modelo básico, é composto por UCP com 64 Kbytes, vídeo de fósforo verde e drive para dois disquetes de 5 1/4", custando 450 ORTN. Na mesma configuração, o JrE custa 480 ORTN. Segundo Ronaldo de Campos Melo, Diretor de Marketing da Itaútec, o Júnior deverá atingir um público formado por pequenas e médias empresas, escolas e profissionais liberais. Com este lançamento, a meta da Itaútec para este ano é de comercializar 64 mil microcomputadores pessoais, atingindo 15% do mercado, e um faturamento de Cr\$ 80 bilhões contra 14 bilhões de 1983.

Brasil já tem light-pen



LP 200,
a primeira
light-pen
brasileira.

A Prisma, uma divisão de produtos da Macromicro, lançou recentemente no mercado brasileiro a LP 200, primeira *light-pen* fabricada no Brasil. Para quem ainda não conhece, a *light-pen* permite o acesso a dados diretamente na tela do monitor, substituindo o teclado nas tarefas de seleção de cores, dados e desenhos, que passam assim a ser feitas diretamente no

visor do microcomputador. Por enquanto apenas os usuários da linha TRS-80 poderão utilizar a LP 200, mas a Prisma promete para breve um modelo específico para a linha Apple. Quaisquer dúvidas sobre como utilizar ou tirar o melhor proveito do produto podem ser esclarecidas pelo telefone (011) 572-3304.

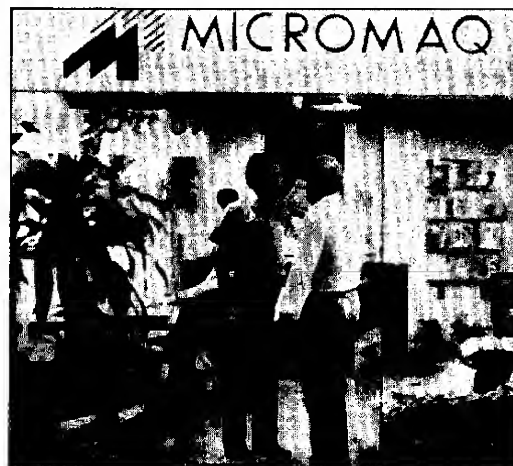
Micromaq investe na área de cursos

A Micromaq, uma das primeiras lojas especializadas em microcomputadores a surgir no Rio de Janeiro, lançou recentemente um curso de Técnicas Digitais e Eletrônica, incrementando o que para a loja é um dos pontos fortes: a área de cursos. Como diz Edson Abib, engenheiro civil e um dos sócios da empresa, "a concorrência é grande e sobrevive quem não se limita a vender software e hardware. É preciso estar sempre na vanguarda e pensar a longo prazo".

Periodicamente a Micromaq oferece cursos de BASIC básico, BASIC avançado, Hardware (Lógica Digital) e Micros para Crianças. Além destes, mantém uma equipe especial, responsável pelos cursos nas empresas. "Estamos também cedendo nosso espaço para reuniões de um clube de usuários do TK82-C e do Color 64", lembra Edson.

Mas não é só no setor de cursos que a loja está investindo. Em sua sede funciona uma oficina especializada capaz de dar assistência técnica a qualquer computador nacional ou estrangeiro, atendendo a pedidos do Brasil inteiro, e foi justamente lá que o Color 64 foi desenvolvido e lançado.

A loja comercializa equipamentos de vários fabricantes nacionais, acompanhados de periféricos para todas as marcas, suprimentos e impressoras. Na parte de software, possui jogos educativos e de animação e desenvolve aplicativos para os TK, Color 64, Microengenho e também para a linha Apple em geral. O endereço da Micromaq é Rua Sete de Setembro, 92, lj. 106, Centro, tel.: (021) 222-6088, Rio de Janeiro, RJ.



Edson Abib: "Sobrevive quem não se limita a vender software e hardware."

Plotter HP para sistemas CAD/CAM

A Hewlett-Packard acaba de lançar um traçador de gráficos para aplicações em projetos de circuitos integrados e impressos, mecânicos, arquitetura e engenharia civil. Na opinião de Brian Moore, Gerente Geral da divisão de San Diego da Hewlett-Packard, "o HP7586B completa a nossa família de produtos para desenho gráfico, atendendo praticamente a todas as exigências dos usuários de CAD/CAM".

O HP7586B trabalha com fo-

lhas de papel tamanhos A4/A até A0/E ou com rolos de papel com largura de 26,7 a 91,9 cm e comprimento de 45,72 m, permitindo assim desenhos contínuos. Possui resolução de 0,025 mm para endereçamento e resolução mecânica de 0,003; sua velocidade é de 60 cm/seg e a aceleração da pena é de 4 g. Compatível com os equipamentos HP, o novo plotter também pode ser usado em alguns computadores de outras marcas.

Clube de manutenção

A M.S. Indústria Eletrônica está lançando um clube de manutenção para possuidores de micros nacionais e importados. Entrando como associado, o usuário terá direito a dois check-ups gratuitos por ano, mão-de-obra gratuita em qualquer conserto ou substituição de peças e assinatura de uma revista técnica, entre outras vantagens. Os sócios do Clube de Manutenção MS receberão trimestralmente uma tabela de descontos para compra de equipamentos em loja especializada em micros e descontos especiais para cursos em escolas credenciadas pelo clube. O Clube M.S. está aberto somente a pessoas físicas, e maiores informações a respeito podem ser obtidas na própria M.S., rua Dr. Astolfo Araújo, 521 - Ibirapuera/São Paulo. Tel.: (011) 549-9022.

STRINGS

★ A *Cobra Computadores e Sistemas Brasileiros S.A.* está iniciando a operação de seu Centro de Computação Gráfica (Computer Aided Design) que permitirá uma significativa redução do ciclo de desenvolvimento de seus produtos. A instalação do Centro visa garantir que a empresa atinja seus objetivos para o biênio 84/85, entre os quais se destaca o desenvolvimento do superminicomputador de tecnologia nacional. ★ Foi inaugurada recentemente na Rua São Januário, 907, em São Cristóvão, no Rio de Janeiro, a *CITEC - Ciência e Tecnologia Livraria e Editora Ltda.*, que inicia suas atividades como distribuidora de livros técnicos e didáticos em geral. Com mais de 500 títulos na área de computação, a CITEC está atendendo pedidos de catálogos através do Serviço de Reembolso Postal ou pelo telefone (021) 580-0034. ★ A *Nashua do Brasil* acaba de lançar a copiadora 3025, que tira 25 cópias por minuto nos formatos A4, ofício e Diário Oficial. O equipamento trabalha simultaneamente com dois cassetes de papel de tamanhos diferentes, tem memória, painel digital e porta-clips magnético. Aceita originais de até 254x356 mm ou maiores, graças à tampa destacável. Seu preço, de Cr\$ 9 milhões e 850 mil, inclui os impostos, e a prestação mensal de leasing situa-se por volta dos Cr\$ 350 mil. ★ A *ADP Systems* está se preparando para ministrar cursos por área de atuação (medicina, engenharia, comunicação, etc.) a partir do segundo semestre deste ano. Estão sendo iniciados também na ADP cursos de pós-graduação de ex-alunos. O primeiro, de uma série voltada a produtos usuais, será o CICS, destinado a aumentar a produtividade. Já na parte de especialização profissional, será desenvolvido Auditoria de Sistemas. ★ Prestando mais um serviço aos usuários da linha TK, a *Microdigital Eletrônica* coloca à disposição o telefone de sua filial para

assistência técnica em Porto Alegre: 24-5060. O endereço é Av. Borges Medeiros, 410, bloco II, 11º andar, conjunto 1108. ★ Já está sendo comercializado o CP500/M, a nova placa CP/M da *Proqus Cibernética Ltda.* O CP/M torna o CP-500 compatível a nível de disco com o Sistema 700, tendo sua capacidade aumentada de 162 K para 190K. ★ A *Informax Assessoria e Comércio de Microcomputadores Ltda.*, que comercializa equipamentos da Prológica, Microdigital e Unitron, mudou-se para a Av. Brigadeiro Faria Lima, 1390/cj. 82-Jd. Paulistano, tel.: (011) 814-0682. ★ A *ABC Bull Telematic* entregou à CTBC, Companhia Telefônica da Borda do Campo, o computador DPS-TL, com 4Mb de memória, 60 terminais de vídeo, quatro impressoras de 100 LPM e quatro unidades de fita magnética. Esses equipamentos executam todos os serviços administrativos da CTBC, e possibilitam a agilização do atendimento aos clientes. ★ A *Tecnocoop Industrial* lançou no mercado sua linha 9000 de impressoras composta pelos modelos TEC 9011 e TEC 9015, com velocidade de 1100 e 1500 LPM respectivamente. A empresa pretende comercializar esse ano 200 máquinas dos novos modelos. Em São Paulo, a Tecnocoop Industrial é representada com exclusividade pela OPT Informática, e no Rio maiores informações sobre as impressoras da linha 9000 podem ser obtidas na própria Tecnocoop, rua do Carmo, 11/4º andar. ★ O *Centro Ótica Comércio e Indústria Ltda.*, tradicional revendedor mineiro de lentes de contato e armação de óculos, inaugurou recentemente um showroom para vendas e demonstrações de micros. Além de hardware, a empresa também comercializará software (inclusive por encomenda), dará orientação técnica e cursos noturnos de linguagem BASIC. Informações pelo telefone (031) 225-2599, ramal 42, Belo Horizonte, MG.

DF incentiva criação de soft

A Companhia de Desenvolvimento do Planalto Central — Codeplan está para implantar, em Brasília, uma software-house pública, que irá permitir aos profissionais de software que não dispõem de equipamentos desenvolver os seus programas, sem nenhum ônus, em máquinas cedidas pelo governo do Distrito Federal.

Para tanto, a Codeplan está instalando um centro de processamento de dados que irá funcionar 24 horas por dia, os sete dias da semana. Os gastos com implantação e manutenção do centro, tais como suprimentos, energia elétrica, pessoal e segurança correrão por conta do governo do Distrito Federal, enquanto que os equipamentos deverão ser fornecidos pelos fabricantes, que também serão responsáveis pela sua instalação e manutenção. Esses equipamentos poderão ser desde microcomputadores até terminais de vídeo ligados a grandes computadores instalados remotamente.

O sistema irá funcionar da seguinte maneira, de acordo com documento divulgado pela Secretaria de Governo do DF: qualquer cidadão do DF, desde que cadastrado na Codeplan atra-

vés da entrega de currículo adequado, poderá apresentar um projeto para o desenvolvimento de software aplicativo. Em prazo nunca superior a 72 horas, a Codeplan julgará se o projeto é aceitável ou não e, em caso afirmativo, indicará o equipamento e o horário que estarão reservados ao interessado, bem como o total de horas em que ele poderá dispor do equipamento (esse total poderá ser ampliado em até 50%, a critério da Codeplan). O software desenvolvido será propriedade de quem o desenvolveu e, ao término da implementação, a Codeplan analisará a sua viabilidade comercial.

A criação da software-house pública faz parte de um esforço que o governo do Distrito Federal vem realizando com vistas a estabelecer em Brasília um pólo de Informática, sobretudo no que diz respeito à indústria de software. No ano passado, dentro do de um plano de modernização administrativa, o governo da capital federal gastou perto de Cr\$ 2 bilhões em produtos e serviços de Informática, sendo que 60% desses recursos destinou-se à contratação de software-houses locais.

Comunicação de dados cresce 90% em 1983

A comunicação de dados foi o segmento que mais cresceu em 1983 dentre os serviços oferecidos pela Embratel: 90%. De acordo com a empresa, os circuitos nacionais faturados, que totalizavam 1 mil 999 em 1982, chegaram a 3 mil 649 em fins do ano passado, atendendo a cerca de 900 empresas espalhadas por mais de 200 localidades do país. Na área internacional, os serviços Airdata, Interdata e Findata eram utilizados, em dezembro de 1983, por mais de 100 usuários. De 1979 a 1983, os serviços de comunicação de dados apresentaram um crescimento anual médio de 179%, informou a Embratel.

DGT-1000 na escola

A Digitus acaba de lançar, para o DGT-1000, uma Unidade Controladora de Terminais Educacionais (UCTE) que permite a comunicação entre um micro central e 16 secundários através da porta de cassete, visando incentivar a utilização de seu novo equipamento em centros educacionais, cursos monitorizados, escritórios etc.

O sistema permite ao instrutor, via computador central, carregar programas nas estações individuais, podendo aplicar provas e obter as respostas automaticamente gravadas nos disquetes de seu computador para posterior avaliação. Nas estações, os digitadores ou estudantes também podem gravar programas diretamente no computador do instrutor e ainda utilizar o seu equipamento independentemente.

A configuração necessária para o DGT-1000 central é de 48 Kb de memória e disk drive; os outros 16 micros podem ser de configuração mínima, ou seja, UCP de 16 Kb e monitor de vídeo.

Ringo com impressora

A partir de agora, qualquer programa desenvolvido no Ringo já pode ser impresso. A Ritao do Brasil acaba de colocar no mercado a interface que possibilita a ligação do Ringo com impressora ou com máquina de escrever elétrica. O Ringo pode também se comunicar com outro computador do mesmo modelo por meio de telefone, através de Modem. E já estão disponíveis alguns cartuchos para o Ringo, entre os quais Órgão, que faz com que o usuário possa compor e executar músicas a

partir do teclado do micro; Cadastro, que mantém na memória do equipamento todos os dados sobre clientes e fornecedores, e o Editor Z 80, para aqueles que quiserem desenvolver seus próprios programas. A Ritao fornece também cartuchos virgens para os usuários que quiserem gravar programas originais em EPROM, entre outras coisas. O Ringo tem uma velocidade de gravação e leitura de fitas de 2400 bps, bastante rápida para um micro de sua faixa.

Nova fábrica, novos periféricos

A Elebra Informática, ao completar cinco anos, apresentou sua nova fábrica de periféricos e comunicou o lançamento de seus novos produtos. São eles: duas unidades de disco magnético Winchester, uma de 340 Mbytes e outra de 515 Mbytes, e uma unidade de fita magnética Streamer, com densidade de 1600 a 6250 BPI. Álvaro Nascimento, Diretor da Elebra, defende a manutenção da reserva de mercado para que os investimentos em pesquisa prossigam. Ele afirmou ainda que, de um faturamento estimado em Cr\$ 150 bilhões, para esse ano, cerca de 25% virão da exportação de placas eletrônicas, vendidas para a empresa americana Control Data. Os diretores da Elebra garantem que no ano que vem os seus produtos já terão preços competitivos no mercado internacional. Hoje a empresa exporta para a América Latina, mas pretende atingir também os mercados americano e europeu.

UNITRON À TODA FORÇA

"Desconfie dos preços muito baixos porque você pode estar comprando material encaixado ou de origem duvidosa." A advertência é do diretor presidente da Unitron, Geraldo Antunes, ao referir-se aos inúmeros modelos compatíveis com o Apple lançados no mercado recentemente. Segundo Geraldo, estes novos equipamentos a preços chamativos atingem principalmente o público formado por pessoas físicas e profissionais liberais, o que vem a firmar ainda mais uma antiga posição da Unitron: dirigir seu produto ao setor profissional.

Dentro desta estratégia, a empresa já está colocando no mercado vários periféricos opcionais, como drive para disquetes de 8", te-

clado profissional acrescido de numérico reduzido e um monitor de vídeo fabricado pela própria Unitron. A empresa está fornecendo também a linha completa de expansões para o seu equipamento, incluindo o cartão controlador de instrumentação, padrão HP, que permite ligar o AP II a aparelhos como analisador de assinaturas e scanner, gerador de varreduras para controle de processos (ambos fabricados pela Hewlett-Packard).

Entre os planos para um futuro próximo, a Unitron pretende aperfeiçoar a área gráfica, e neste sentido já estão sendo desenvolvidos projetos de vídeo a cores, software específico para este setor e prancheta digitalizadora. A área de CAD é um outro setor onde a Unitron

está investindo, visando a obtenção de um acessório barato para desenho técnico e desenvolvimento de circuitos impressos, entre outras aplicações.

Disponibilidade de software para o seu equipamento é outra preocupação da Unitron, que no momento está credenciando software houses com atenção voltada principalmente para as áreas de transporte, educação, saúde e hotelaria. "Nossa meta", explica Geraldo Antunes, "é prover de software para as várias áreas de aplicação todos os pontos de venda do AP II. Este é o melhor investimento de marketing que podemos fazer — software a baixo custo para o usuário", conclui.

Informática em Ribeirão Preto

Com a finalidade de oferecer às empresas de computação a oportunidade de exporem, com vendas diretas, seus produtos e serviços no interior de São Paulo, o SENAC de Ribeirão Preto realizou, de 25 a 28 de abril, a I Feira de Informática da região. A iniciativa objetivou ainda proporcionar ao público consumidor e em geral a chance de conhecer, analisar e avaliar os produtos e serviços oferecidos pelo atual mercado nacional de Informática.

Várias empresas estiveram presentes na Feira, entre elas a Prológica, Microdigital (que aproveitou a ocasião para lançar no interior do Estado o seu TK2000), Dismac, Digitus, Polymax e Unitrone. A grande atra-

ção, no entanto, ficou por conta da Memocards, empresa de Ribeirão Preto especializada na revenda de microcomputadores e periféricos que vêm se destacando na produção de software para as áreas comercial, de Engenharia Civil e, principalmente, educacional.

Em seu estande, o maior da feira, a Memocards demonstrou o trabalho que vem desenvolvendo em conjunto com os alunos do Colégio Anglo, o qual tem resultado em uma série de programas específicos para a área educacional, tais como: *Mapa do Brasil, Características dos Animais, Verbos e Estórias em Inglês, Introdução às Crises e Resumo de Vestibulares para Microcomputadores*. Com

excessão deste último, todos os demais programas desta área não são comercializados.

Paralelamente à Feira, realizou-se também o II Simpósio de Informática, onde foram discutidos temas ligados à evolução e perspectivas da Informática no Brasil. Na oportunidade, Ademir Potiens, gerente do SENAC e organizador da Feira, assegurou que, sendo Ribeirão Preto um importante parque comercial dentro do Estado, não pode ficar alheio aos progressos da Informática, e declarou: "Várias empresas da região já estão na era da computação, mas ainda há as que precisam melhorar o seu desempenho; esta é a nossa principal preocupação ao promovermos eventos desta natureza".

Craft II, o Apple da Microcraft

Nos primeiros dias de abril os jornais de São Paulo anunciavam que um novo microcomputador da linha Apple estava sendo lançado a um preço especial de Cr\$ 1 milhão e 90 mil, com exclusividade no Mappin, um dos maiores magazines da cidade. Tratava-se do Craft II, da Microcraft, empresa já conhecida na área de Informática pela fabricação de placas para microcomputadores da linha Apple.

A escolha do Mappin, por seu poder de penetração junto ao público consumidor, teve resposta imediata: a previsão de vendas de 150 equipamentos no primeiro mês estourou nas duas primeiras semanas, e a partir do mês de maio o micro já estará sendo vendido também em lojas especializadas.

Segundo Jae Yoon, diretor da Microcraft, o objetivo da empresa ao lançar o Craft II foi oferecer um Apple profissional a um preço intermediário entre esta faixa e a faixa dos pessoais. Sua configuração básica é composta por UCP com microprocessador 6502, 48 Kb de memória RAM, 12 Kb de ROM, teclado alfanumérico com auto repetição e oito slots de expansão para conexão de cartões periféricos (todos os cartões fabricados pela Microcraft podem ser conectados diretamente no Craft II). O novo micro tem saída para gravador cassete, opção para trabalhar com disquetes, e o usuário poderá utilizar uma televisão comum (preto e branco ou colorida) ou monitor de vídeo. O Craft II é totalmente compatível, tanto em hardware quanto em software, com o Apple II e seus similares nacionais.

Computação no escritório

As últimas novidades em equipamentos, utensílios e serviços de escritório, inclusive na área de processamento de dados e teleprocessamento, com tecnologia nacional ou estrangeira, serão apresentados na nova FUSE — Feira Internacional de Utensílios e Serviços de Escritório, que se realizará em São Paulo entre 20 e 24 de junho próximo.

Patrocinada pela Federação Nacional dos Bancos e pela Federação Brasileira de Bancos, a FUSE estará montada no pavilhão de exposições do Parque Anhembi e, além do setor de automação de escritórios, estará dividida em comunicação e telecomunicação; arquivamento, classificação e microfilingagem; instalação e mobiliário; máquinas de escrever, cálculo, contabilidade, processamento de dinheiro e cheques, acessórios e materiais; controles visuais e relógios; segurança e controle; artigos de papelaria, desenho e escrita; consultoria e assessoria; brindes; serviços em geral; publicações; ensino e pesquisa; bancos.

A Feira funcionará no horário comercial, diariamente, inclusive aos sábados e domingos.

Convênio SENAI-CTI

O SENAI SP e o Centro Tecnológico para Informática assinaram um convênio para a realização de atividades conjuntas de pesquisa, desenvolvimento e produção de máquinas e equipamentos didáticos. Esse material será utilizado no ensino de Informática e Automação Industrial, como parte de projetos destinados à formação pelo SENAI de recursos humanos para indústrias do setor de eletrônica. A primeira atividade conjunta já está em andamento: trata-se do treinamento de técnicos do SENAI paulista, docentes nas áreas de eletrônica, mecânica de precisão e instrumentação, além de técnicos em supervisão de ensino e material didático, nas áreas de robótica industrial e sistemas operacionais para microcomputadores. Após o treinamento, esses técnicos vão desenvolver projetos de construção de equipamentos didáticos de automação industrial e de equipamentos didáticos para o ensino de informática.

Microtecla

Foi inaugurada recentemente em Franca, interior de São Paulo, a Microtecla, loja especializada na comercialização de microcomputadores pessoais e profissionais. Na nova loja podem ser encontrados equipamentos da Labo, Prológica, Dismac, Unitrone e Microdigital, além de suprimentos e livros e revistas da área. No departamento denominado Boutique, encontram-se calculadoras, fitas, jogos, relógios-game e videogames. A Microtecla fica na Rua Major Caludiano, nº 1.445, Franca, São Paulo, tel.: (016) 722-2820.

MS É FEITA PARA VOCÊ PARTICIPE COM SUA OPINIÃO



Escreva-nos dizendo qual a sua área de interesse, conte-nos também as suas experiências com seu micro, o que você quer ver

publicado em MS, o que você acha da sua MS, enfim, diga tudo que tome MICRO SISTEMAS ainda mais feita para você.

E lembre-se: todo leitor que nos escreve concorre automaticamente a uma assinatura de um ano de nossa MICRO SISTEMAS. Mande logo sua opinião para Redação de MICRO SISTEMAS no Rio de Janeiro ou em São Paulo: Av. Presidente Wilson, 165/grupo 1210 - Centro - CEP 20030 - Rio de Janeiro - RJ; Rua Oliveira Dias, 153 - Jardim Paulista - CEP 01433 - São Paulo - SP.

Antes de aplicar seu rico dinheirinho, calcule, com seu micro da linha Sinclair, os juros, o montante ou o capital de investimento.

Para bom investimento, um micro pessoal basta

Armando Oscar Cavanha Filho

Este programa, elaborado em um TK82-C, tem a finalidade de calcular algumas variáveis financeiras a partir de valores conhecidos. As variáveis previstas no programa são:

- . P — principal, ou seja, o capital no dia de hoje (capital no início da contagem dos períodos de capitalização);
 - . S — montante, ou seja, o capital no final do período (capital no final da contagem dos períodos de capitalização);
 - . N — número de períodos de capitalização;
 - . I — taxa de juros por período de capitalização;
 - . J — taxa de juros compostos em N períodos de capitalização;
 - . R — série uniforme de pagamentos, ou seja, série uniforme de valores de retirada ou entrada, sendo cada parcela entendida como no final de cada período.
- O programa funciona de acordo com a tabela apresentada na figura 1.

COMO PROCEDER

Veja agora, através de dois exemplos, como funciona o programa.

1 — Quanto teremos daqui a 12 meses se aplicarmos hoje Cr\$ 1.000,00 a 2,5% ao mês?

- a) São dados: N = 12, P = 1000,00 e I = 2,5; deseja-se calcular S.
b) Digita-se o código que representa o problema (no caso, achar S a partir de N, P e I, ou seja, código 0).

DADOS	CALCULA	CÓDIGO
P, I, N	S	0
S, I, N	P	1
S, P, I	N	2
S, P, N	I	3
P, I, N	R	4
R, I, N	P	5
R, P, I	N	6
R, P, N	I	7
R, I, N	S	8
S, I, N	R	9
R, S, I	N	10
J, N	I	11

Figura 1

c) Introduzem-se os valores na ordem pedida.

d) Obtém-se a resposta: S = 1344,88.

2 — Um vendedor de carros apresenta um Volks usado por Cr\$ 100 mil de entrada, mais 12 prestações de Cr\$ 50 mil. O preço à vista é de Cr\$ 350 mil. Qual a taxa de juros do financiamento?

a) São dados: P = 350.000 - 100.000

= 250.000, N = 12 e R = 50.000; deseja-se calcular I.

b) Digita-se 7.

c) Introduzem-se os dados.

d) Resposta: I = 16,94%.

OUTROS EXEMPLOS

3 — Quanto devo pagar hoje para ter o direito de receber Cr\$ 10 mil daqui a cinco anos, a juros de 10% ao ano?

S = 10.000

N = 5

I = 10

Resposta: P = 6209,21

4 — Um empresário pretende fazer um investimento que renderá Cr\$ 100 mil por ano, nos próximos 10 anos. Qual o valor do investimento, sabendo que a taxa de juros é de 6% ao ano?

R = 100.000

N = 10

I = 6% aa

Resposta: P = 736.008,71

5 — Quanto devemos depositar semestralmente numa conta a prazo fixo, que paga juros de 12% por semestre, para termos Cr\$ 50 mil daqui a sete anos?

I = 12% ao semestre

S = 50.000

N = 7 x 2 = 14 semestres

Resposta: R = 1543,56

6 — Quais os juros mensais equivalentes a 34,4% ao ano?

J = 34,4% aa

N = 12 meses

Resposta: I = 2,49% ao mês

Cálculo de Parâmetros Financeiros

```

10 REM "FINANCEIRO"
15 CLS
17 PRINT
20 PRINT "ESTE PROGRAMA CALCUL
A ALGUNS"
30 PRINT "PARAMETROS FINANCEIR
OS : "
40 PRINT
60 PRINT " DADOS          CALCULA
CODIGO"
70 PRINT " P,I,N-----S-----
-----0"
80 PRINT " S,I,N-----P-----
-----1"
90 PRINT " S,P,I-----N-----
-----2"
100 PRINT " S,P,N-----I-----
-----3"
110 PRINT " P,I,N-----R-----
-----4"
120 PRINT " R,I,N-----P-----
-----5"
130 PRINT " R,P,I-----N-----
-----6"
140 PRINT " R,P,N-----I-----
-----7"
150 PRINT " R,I,N-----S-----
-----8"
160 PRINT " S,I,N-----R-----
-----9"
170 PRINT " R,S,I-----N-----
-----10"
180 PRINT " J,N-----I-----
-----11"
190 PRINT
200 PRINT "PARA INICIAR, INTROD

UZA O NUMERO DE CODIGO E NEWLINE
"
210 INPUT COD
220 CLS
230 IF COD=0 THEN GOTO 360
240 IF COD=1 THEN GOTO 480
250 IF COD=2 THEN GOTO 600
260 IF COD=3 THEN GOTO 720
270 IF COD=4 THEN GOTO 840
280 IF COD=5 THEN GOTO 960
290 IF COD=6 THEN GOTO 1080
300 IF COD=7 THEN GOTO 1200
310 IF COD=8 THEN GOTO 1410
320 IF COD=9 THEN GOTO 1530
330 IF COD=10 THEN GOTO 1650
340 IF COD=11 THEN GOTO 1770
350 IF COD>11 THEN GOTO 1870
360 PRINT "INTRODUZA P"
365 INPUT P
370 PRINT "INTRODUZA I"
375 INPUT I
380 PRINT "INTRODUZA N"
390 INPUT N
410 CLS
420 LET S=P*(1+I/100)**N
430 PRINT "S=";S
435 PRINT
440 PRINT "P=";P
445 PRINT
450 PRINT "I=";I
455 PRINT
460 PRINT "N=";N
470 GOTO 1890
480 PRINT "INTRODUZA S"
485 INPUT S
490 PRINT "INTRODUZA I"

495 INPUT I
500 PRINT "INTRODUZA N"
505 INPUT N
530 CLS
540 LET P=S/(1+I/100)**N
550 PRINT "P=";P
555 PRINT
560 PRINT "S=";S
565 PRINT
570 PRINT "I=";I
575 PRINT
580 PRINT "N=";N
590 GOTO 1890
600 PRINT "INTRODUZA S"
605 INPUT S
610 PRINT "INTRODUZA P"
615 INPUT P
620 PRINT "INTRODUZA I"
625-INPUT I
650 CLS
660 LET N=(LN (S/P))/(LN (1+I/1
00))
670 PRINT "N=";N
675 PRINT
680 PRINT "S=";S
685 PRINT
690 PRINT "P=";P
695 PRINT
700 PRINT "I=";I
710 GOTO 1890
720 PRINT "INTRODUZA S"
725 INPUT S
730 PRINT "INTRODUZA P"
735 INPUT P
740 PRINT "INTRODUZA N"
745 INPUT N

```

PUBLICAÇÕES PARA COMPUTADORES

sinclair®

SINCLAIR É MARCA REGISTRADA DE SINCLAIR RESEARCH LTD

• APLICAÇÕES SÉRIAS PARA TK 85 E CP 200

3ª Edição, atualizada e com nova composição gráfica

CR\$ 9.000

Quem é Sinclair? - Convertendo outros Basics - Contando os Bytes - Economizando Memória - Fluxogramas - Top Down - Erros da ROM - Conhecendo a Impressora - Chaining Programas - Sub-rotinas em Cassete - Folha de Pagamento - Balancete - Correção Monetária do Imobilizado - Das Contribuições do IAPAS - Contas a Receber - Cadastro de Clientes - Conta Bancária - Correção de Provas - Processador de Textos - Estatística - Custos - Orçamento Doméstico - Ram Top e Data File em Código - etc.

• 45 PROGRAMAS PRONTOS PARA RODAR EM TK 82C E NE Z800

8ª Edição, reimpressa

CR\$ 5.500

Arquivos - Estoque - Plano Contábil - Folha de Pagamento - Agenda Telefônica - Caça ao Pato - Trilha - Jogo da Velha - Forca - Dado - Tabelas - Tabuada - Conversão de Coordenadas - Média - Progressão - Tabela Price - Fibonacci - Depreciação - Renumerador de linhas em Código - etc...

• 30 JOGOS PARA TK 82C E CP 200

3ª Edição

CR\$ 6.000

Damas - Labirinto - Enterprize - Golfe - Velha - Visita ao Castelo Cassino - Roleta Russa - Corrida de Cavalos - Vinte e Um - Cubo Mágico - Senha - Banco Imobiliário - Forca - Dados - Invasores - etc.

PROGRAMAS NO CÓDIGO DA MÁQUINA

Inversão de Vídeo - Som por Software - Labirinto - Destrava Soft

• CÓDIGO DE MÁQUINA PARA TK E CP 200

1ª Edição

CR\$ 9.000

Números Binários e Hexadecimais - Arquitetura do Z80 - Editando em Código - Programa para Edição - As Instruções do Z80 em Exemplos Sub-rotinas da ROM - A ROM de 8K - Dicionário das Instruções - Hex X Mnemônicos - Hex X Decimal - Incluindo os Programas Scroll - Save Display no Ram Top - Contadores de Pontos ou Tempo - Datafile - Renumber - Labirinto - som por Software - Micron Pac - Bombardeio etc...

Despachos para todos os Estados mediante Ordem de Pagamento ou Cheque Nominal

MICRON ELETRÔNICA COM. IND. LTDA.

Av. São João, 74 - Tel. (0123) 22-4194 - cep 12.200 - S. J. Campos-SP

PARA BOM INVESTIMENTO, UM MICRO PESSOAL BASTA

```

770 CLS
780 LET I=((S/P)**(1/N))-1
790 PRINT "I=";I*100
795 PRINT
800 PRINT "S=";S
805 PRINT
810 PRINT "P=";P
815 PRINT
820 PRINT "N=";N
830 GOTO 1890
840 PRINT "INTRODUZA P"
845 INPUT P
860 PRINT "INTRODUZA I"
870 INPUT I
875 PRINT "INTRODUZA N"
880 INPUT N
890 CLS
900 LET R=P*((I/100)/(1-(1+I/100)**(-N)))
910 PRINT "R=";R
915 PRINT
920 PRINT "P=";P
925 PRINT
930 PRINT "I=";I
935 PRINT
940 PRINT "N=";N
950 GOTO 1890
960 PRINT "INTRODUZA R"
980 INPUT R
985 PRINT "INTRODUZA I"
990 INPUT I
995 PRINT "INTRODUZA N"
1000 INPUT N
1010 CLS
1020 LET P=R*((I/100)/(1-(1+I/100)**(-N)))**(-1))
1030 PRINT "P=";P
1035 PRINT
1040 PRINT "R=";R

```

```

1045 PRINT
1050 PRINT "I=";I
1055 PRINT
1060 PRINT "N=";N
1065 GOTO 1890
1080 PRINT "INTRODUZA I"
1100 INPUT I
1105 PRINT "INTRODUZA P"
1110 INPUT P
1115 PRINT "INTRODUZA R"
1120 INPUT R
1130 CLS
1132 IF R<P*I/100 THEN GOTO 2340
1140 LET N=(LN (R/(R-P*I/100)))/(LN (1+I/100))
1150 PRINT "N=";N
1155 PRINT
1160 PRINT "I=";I
1165 PRINT
1170 PRINT "P=";P
1175 PRINT
1180 PRINT "R=";R
1190 GOTO 1890
1200 PRINT "INTRODUZA R"
1220 INPUT R
1225 PRINT "INTRODUZA P"
1230 INPUT P
1235 PRINT "INTRODUZA N"
1240 INPUT N
1250 CLS
1255 PRINT "SE O PROCESSAMENTO D
EMORAR"
1256 PRINT "MAIS QUE 2 MIN CONSU
LTAR O"
1257 PRINT "AUTOR DO PROGRAMA"
1258 PRINT "VIA MICRO SISTEMAS"
1260 LET Q=R*N
1270 IF Q>P THEN GOTO 1300
1275 CLS
1280 PRINT "IMPOSSIVEL; REVEJA S
EUS DADOS"
1285 PRINT "PORQUE N*R TEM QUE S
ER >P"
1290 GOTO 17
1300 LET IO=((R*N/P)**(1/(N+1)))-1
1310 LET RO=(P*IO)/(1-(1+IO)**(-N))
1320 LET Y=ABS (R-RO)
1330 IF Y>0.1 THEN GOTO 1390
1335 CLS
1340 PRINT "I=";IO*100
1345 PRINT
1350 PRINT "R=";R
1355 PRINT
1360 PRINT "P=";P
1365 PRINT
1370 PRINT "N=";N
1380 GOTO 1890
1390 LET IO=IO+((R-RO)/R)*IO
1400 GOTO 1310
1410 PRINT "INTRODUZA R"
1430 INPUT R
1435 PRINT "INTRODUZA I"
1440 INPUT I
1445 PRINT "INTRODUZA N"
1450 INPUT N
1460 CLS
1470 LET S=R*((1+I/100)**N-1)/(I/100)
1480 PRINT "S=";S
1485 PRINT
1490 PRINT "R=";R
1495 PRINT
1500 PRINT "I=";I
1505 PRINT
1510 PRINT "N=";N
1520 GOTO 1890
1530 PRINT "INTRODUZA S"
1550 INPUT S
1555 PRINT "INTRODUZA I"
1560 INPUT I

```

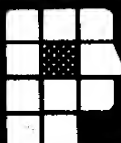
```

1565 PRINT "INTRODUZA N"
1570 INPUT N
1580 CLS
1590 LET R=S*((1+I/100)**N-1)/(I/100)**(-1)
1600 PRINT "R=";R
1605 PRINT
1610 PRINT "S=";S
1615 PRINT
1620 PRINT "I=";I
1625 PRINT
1630 PRINT "N=";N
1640 GOTO 1890
1650 PRINT "INTRODUZA R"
1670 INPUT R
1675 PRINT "INTRODUZA S"
1680 INPUT S
1685 PRINT "INTRODUZA I"
1690 INPUT I
1700 CLS
1710 LET N=(LN ((S*I/(100*R))+1))/(LN (1+I/100))
1720 PRINT "N=";N
1725 PRINT
1730 PRINT "R=";R
1735 PRINT
1740 PRINT "S=";S
1745 PRINT
1750 PRINT "I=";I
1760 GOTO 1890
1770 PRINT "INTRODUZA J"
1790 INPUT J
1795 PRINT "INTRODUZA N"
1800 INPUT N
1810 CLS
1820 LET I=((1+J/100)**(1/N)-1)*100
1830 PRINT "I=";I
1835 PRINT
1840 PRINT "J=";J
1845 PRINT
1850 PRINT "N=";N
1860 GOTO 1890
1870 CLS
1875 PRINT "CODIGO =" ;COD;" ,INEX
ISTENTE."
1880 GOTO 17
1890 FOR M=10 TO 50
1900 PLOT M,15
1910 NEXT M
1920 FOR K=15 TO 25
1930 PLOT 10,K
1940 NEXT K
1950 FOR L=15 TO 30
1960 PLOT 50,L
1970 NEXT L
1980 FOR Q=10 TO 15
1990 PLOT 13,Q
2000 PLOT 16,Q
2010 PLOT 19,Q
2020 PLOT 22,Q
2030 PLOT 25,Q
2040 PLOT 47,Q
2050 PLOT 50,Q
2055 NEXT Q
2060 PRINT AT 10,3;"P"
2070 PRINT AT 7,27;"S"
2080 PRINT AT 16,14;"SERIE""R""
2090 PRINT AT 9,8;"I=JUROS/PERIO
DO"
2100 PRINT AT 11,8;"N=N DE PERI
ODOS"
2120 PRINT AT 14,26;"TEMPO"
2300 PRINT AT 20,0;"PARA CONTINU
AR DIGITE 1"
2310 INPUT W
2320 IF W=1 THEN GOTO 15
2330 STOP
2340 PRINT "ATENCAO, ERRO : R>P*
I/100 SEMPRE"
2350 GOTO 17

```

Tecnologias
* **Microcomputadores**
* **Suprimentos**
* **Software**
* **Bureau de Serviço**
* **Consultoria**
* **Auditoria de Sistemas**

Av. do Contorno, 5826 /3º andar
— Savassi — Belo Horizonte —
PABX (031) — 223-6000



PROLOGICA
microcomputadores

REVENDEDOR AUTORIZADO

Armando Oscar Cavanha Filho é Engenheiro Mecânico formado pela Universidade Federal do Paraná.

RINGO R-470 CHEGOU À CIDADE. OS OUTROS MICROS QUE SE CUIDEM.



Cr\$ 449.950,00

O microcomputador Ringo R-470 é, disparado, o melhor em sua categoria. É mais rápido na execução de programas, oferece amplas possibilidades de expansão e é o único micro totalmente projetado e desenvolvido no Brasil, aprovado pela SEI - Secretaria Especial de Informática.

Você pode contar com ele para resolver problemas pessoais ou profissionais, ou simplesmente para se divertir, através de vários jogos disponíveis em cartuchos ou fitas.

Aliás, cartucho é o que não falta para o Ringo. Ele é compatível com todos os programas do famoso Sinclair e possui equipamentos periféricos exclusivos que ampliam muito a sua capacidade.

Compare o Ringo R-470 com os similares e comprove: nunca apareceu um micro pessoal tão profissional por estas redondezas.

CARACTERÍSTICAS TÉCNICAS:

- Linguagem Basic e códigos de máquina Z-80
- 8 KBytes ROM expandível para 16 K Bytes
- 16 KBytes RAM expandível para 48 K Bytes
- Utilizável em qualquer TV P&B ou cores
- Conector para Joystick (jogos)
- Teclado tipo QWERTY com 49 teclas e 155 funções - teclas de edição (movimentação de cursor e correção) com repetição automática
- Exclusiva tecla de inversão de vídeo
- Tela com 24 linhas de 32 colunas para texto

- Resolução gráfica 64 x 44 pixels (unidade gráfica), podendo atingir uma matriz de 256 x 192 quando utilizado com cartuchos
- Cálculos aritméticos, funções trigonométricas, logarítmicas e lógicas
- Cartuchos "Instant Soft" (programas aplicativos em ROM - exclusivo)
- Velocidade de gravação em fita cassete 2.400 BPS

EXPANSÕES:

- Gravador de EPROM para gravar, editar e copiar programas em cartucho
- Interface para impressora ou máquina de escrever elétrica
- Sintetizador de sons
- MODEM (Comunicação telefônica - 1.200 Bauds)

* Preço sujeito a alteração

À venda nas lojas especializadas em micros, foto-vídeo-som e grandes magazines. Não encontrando o Ringo nestes locais, ligue para 217.8400 (SP) ou (011) 800.8441 e 800.8442 (Outras localidades do Brasil). DDD gratuito.

RINGO R-470

O micro que aceita desafios.

Ritas do Brasil Ltda. - Divisão Informática
Telex (011) 34673 Rita BR



Enxadrista experiente, Luciano Nilo de Andrade já escreveu para os jornais "Correio da Manhã", "Data News" e "Última Hora" e para a revista "Fatos & Fotos". Luciano é economista, trabalhando no Ministério da Fazenda, no Rio de Janeiro. As opiniões e comentários de Luciano Nilo de Andrade, bem como as últimas novidades do Xadrez jogado por computadores, estarão sempre presentes em MICRO SISTEMAS.

Qual o melhor micro?

Esta é uma pergunta que muitos leitores me fazem, difícil de responder dada a crescente evolução tecnológica da microinformática que a cada dia nos surpreende com equipamentos dotados de recursos mais e mais apurados. Mas podemos, hoje, citar o *Elite A/S*, o qual venceu o III Campeonato Mundial de Micros realizado no final do ano passado, cujos resultados já foram divulgados nesta seção (há que considerar que o emparelhamento adotado foi o suíço, o que dá azo ao fator sorte).

Em testes realizados aqui no Rio, ele ligeiramente se impôs por jogar melhor as aberturas, graças à sua maior memória e leve superioridade na fase final. Em confronto com o *Constellation*, aparelho que custa metade do preço do *Elite*, o escore continua empatado depois de jogadas quatro partidas. Duas relâmpago e duas pensadas. Estas duas últimas vão a seguir reproduzidas para que o leitor possa fazer seu próprio julgamento.

Elite A/S x Constellation Rio, março de 1984

1 - C3BR P4D; 2 - P4BD P5D; 3 - P3CR P4BD; 4 - B2C C3BD; 5 - 0-0 C3B; 6 - P3D P4R; 7 - D4T B2D; 8 - B2D C5CR; 9 - P3TR C3B; 10 - C3T B2R; 11 - D3C D1B; 12 - CxP(?!). Uma ousada entrega de duas peças por uma torre e dois peões. 12 - ...CxC; 13 - BxP D1D; 14 - BxT DxB; 15 - C5C 0-0; 16 - R2T P3TD; 17 - P4B C(4)5C+; 18 - PxC CxP+; 19 - R1C PxC. O *Elite*, afinal, não conseguiu mostrar o que queria com o sacrifício do cavalo na 12ª jogada. 20 - PxP B3R; 21 - D3T D4D!; 22 - TD1R T1T; 23 - B5T(?) D4T; 24 - T2B B1D; 25 - BxB TxD; 26 - PxT CxT; 27 - Rx C D7T+; 28 - R3B B4D+; 29 - P4R D7D; 30 - PxB DxD; 31 - P6D D6R+; 32 - R4C P4T+ e as brancas abandonaram porque

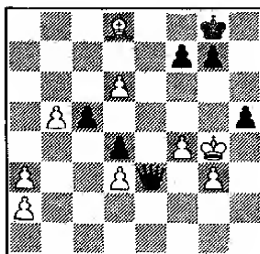


Diagrama A - Posição depois de 32 - R4C P4T+! As brancas jogaram 33 - RxP DxPC; 34 - P7D R2T; 35 - P5B P3C+; 36 - PxC PxC mate. Observação: cada micro dispunha de duas horas para cada 40 jogadas.

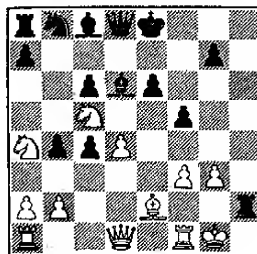


Diagrama B - Posição depois de 16 - 0-0(?) TxP! Observação: cada micro dispunha de duas horas para cada 40 jogadas.

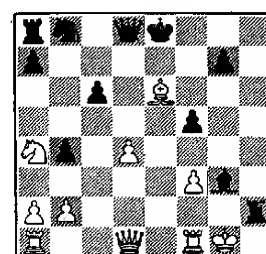


Diagrama C - Posição depois de 19 - BxB. As pretas anunciaram mate em quatro jogadas começando com 19 - ...D5T!; 20 - B7B+ R1B; 21 - T1R T8T; 22 - R2C D6T mate!

previram que levariam mate em quatro jogadas (veja o diagrama A).

Constellation x Elite A/S Rio, março de 1984

1 - P4D P4D; 2 - P4BD P3R; 3 - C3BR C3BR; 4 - C3B P3B; 5 - B5C PxP; 6 - P4R P4CD; 7 - P5R P3TR; 8 - PxC. Comentando esta continuação, há 30 anos, S. Tartakower declarou ser esta uma continuação *unpromising* depois da seguinte sequência: 8 - ...PxP; 9 - PxP BxP etc. O *Constellation* não pensa assim. Bom, o resultado logo sabemos. 8 - ...PxP; 9 - C(3BR)xP DxB; 10 - C(5)4R D1D; 11 - B2R P5C; 12 - C4T T5T; 13 - P3B P4BR; 14 - P3CR T3T; 15 - CR5B B3D; 16 - 0-0(?) TxP! e as pretas fragmentam a estrutura de peões que protegia o monarca branco (veja o diagrama B).

17 - CxPR(?) (as brancas estão desorientadas). Maior resistência ofereceriam com P4B, o que já deveriam ter feito no seu 16º lance. 17 - ...BxC; 18 - BxP BxP! As pretas rejeitam o ganho fácil

do bispo para ameaçarem mate. 19 - BxB e as pretas (o *Elite*) anunciam mate em quatro jogadas (veja o diagrama C).

CONTINUE JULGANDO

Abente x Netto Paraguai, 1983

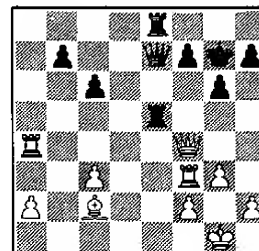


Diagrama D - As pretas jogam e dão mate em seis jogadas. O *Elite* gastou 4m 4s para encontrar a resposta e o *Super 9* precisou de 6m 8s.

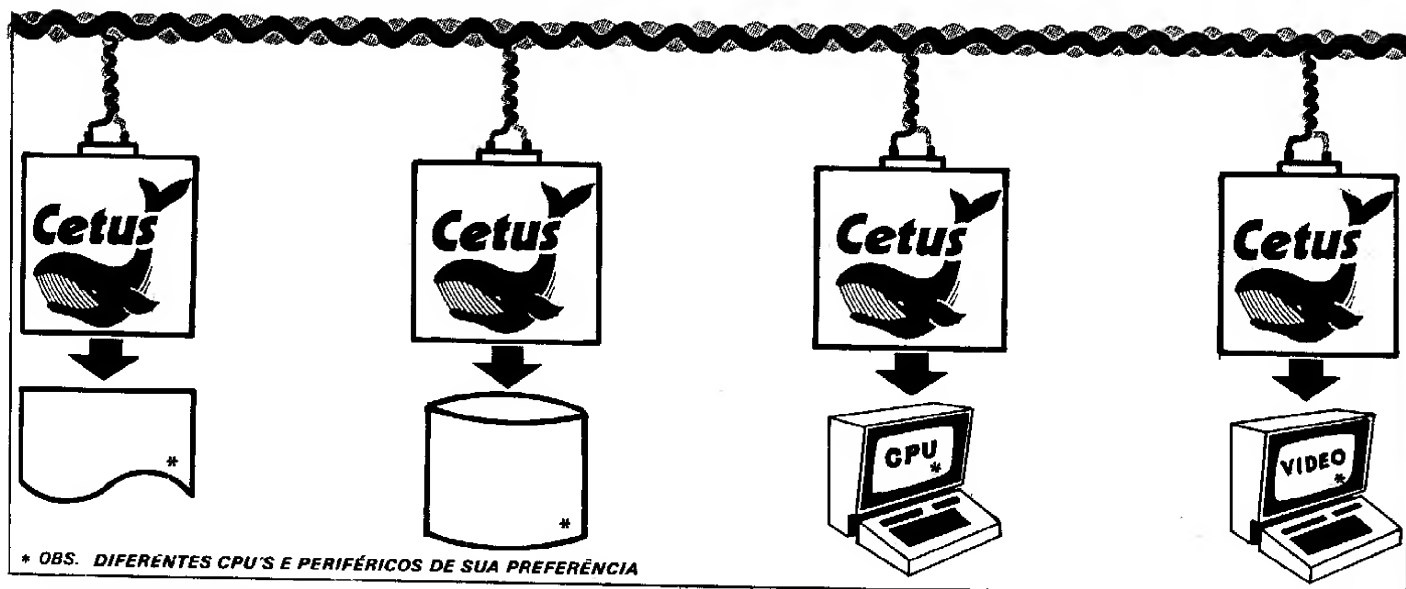
Solução do problema

B6T+; 6 - T8R mate!
3 - HxT D8R+; 4 - R2C D8CR+; 5 - RxD
Diagrama D - 1 - ...T8R+; 2 - R2C T8CR+;

LIGUE-SE NA CETUS



REDES LOCAIS



* OBS. DIFERENTES CPU'S E PERIFÉRICOS DE SUA PREFERÊNCIA

INÉDITO NO BRASIL

A Cetus lança o CS-1000. Um nodo capaz de alimentar a vontade dos seus computadores realizando as funções de comunicação indispensável e minimizando os custos operacionais. O CS-1000 é uma dessas grandes idéias que chegam para ficar.

O QUE É O CS-1000

Como pioneira na implantação de redes locais, a Cetus traz o CS-1000 para permitir a expansão ilimitada do processamento, antes realizado por um grande computador de capacidade limitada.

Ele permite a interligação de computadores, microcomputadores e periféricos, sem nenhum controlador centralizador de rede, fazendo com que as operações dos componentes na rede não sejam prejudicadas pela falha de um ou mais elementos.

Os nodos CS-1000 têm, na função de interface, a capacidade de realizar os protocolos da rede, método de acesso, integridade da comunicação, endereçamento, transferência de dados, empacotamento de mensagens, controle de fluxo, temporizações, detecção e recuperação em caso de falhas, liberando os equipamentos usuários dessas tarefas.

O INÍCIO DA REDE PODE SER UM MICRO

Para capacitar a rede em função da necessidade, os equipamentos adquiridos são progressivamente interligados. O CS-1000 é um elemento inteligente que, ligado à rede, permite comunicação simultânea e independente entre todos os elementos, dois a dois no mínimo. Sua Rede Local vai crescendo a medida das suas necessidades até 255 elementos (computadores, impressoras, terminais, discos, etc.) que se interligam através de um simples par de fios trançados.

REDE LOCAL CETUS

A partir de agora, com a Rede Local Cetus e o CS-1000, o seu sistema vai evoluir tanto, mas tanto, que o processamento no Brasil vai passar a ter duas datas. Uma, anterior, ao CS-1000. A outra, após o surgimento da Rede Local Cetus.

FILIADA A ABICOMP

Rua Alte. Cochrane, 206
Tijuca - Rio de Janeiro
Tels.: (021) 284-7075 ou
284-6659



Informática SA

Esta dica de programação vai para os usuários da linha TRS-80: como reservar espaço na memória para carregar rotinas em linguagem de máquina

Abrindo espaço para a linguagem de máquina

José Ricardo Flores Rodrigues

Vários usuários de micros da linha TRS-80 encontram certa dificuldade quando precisam reservar espaço na memória para carregar um programa em linguagem de máquina. A idéia corrente é que, se o computador já estiver ligado, terá que ser desligado quando surgir **PROTEGER** para reservar a área. Na verdade, porém, isto não é necessário: basta utilizarmos a função **VARPTR** - *Variable Pointer Function*.

VARPTR nos permite encontrar a localização do índice de qualquer variável. A partir daí, mandamos o computador calcular a sua localização e armazenar

nossa rotina em linguagem de máquina dentro desta variável.

Digite o programa que está na listagem 1 para ver como funciona. Feito isso, digite **RUN** e entre com os seguintes códigos:

```
128 156 128 136 148 251
168 159 189 159 189 251
130 171 135 175 131 251
128 176 187 177 144 251
128 160 151 181 128 128
```

Você acabou de colocar caracteres gráficos dentro da *string* **ST\$**. O código 251 imprime 59 espaços em branco. Dê um **PRINT ST\$** e logo após um **LIST 1**. Notou que coisa *horrivelmente* interessante ficou a variável **ST\$**? Para evitar que o conteúdo do **ST\$** fique desta forma, utilize **ST\$ = STRING\$(30,0)**; e antes de continuar, entre com o seguinte: **FOR A = 0 TO 800 STEP 70:CLS:PRINT @ A,ST\$;NEXT**.

A partir do momento em que a rotina em linguagem de máquina estiver armazenada na memória, executá-la num programa escrito em BASIC é bem simples se utilizarmos a função **USR**. Existem, entretanto, alguns detalhes que você já sabe mas que seu computador precisa ser informado através do programa.

Você precisa informar, por exemplo, se está trabalhando com disco ou fita. Experimente usar:

```
CLS: IF PEEK (16396) =
201 PRINT "NAO ESTOU
COM DISCO! E VOCE SA
BE"
ELSE "CLARO QUE SIM
SEU TOLO"
```

Se você não possui sistema de disco, use o **POKE** nas posições 16526 e 16527; caso contrário, utilize a função **DEFUSR = L**. Resumindo:

```
IF PEEK (16396) = 201
POKE 16526, LS: POKE
16527, MS ELSE DEFUSR
=L
```

É importante também trabalhar sempre em localizações de memória compatíveis com a capacidade de seu computador, ou seja, **IF L > 32767 L = L - 65536**. Além disso, a constante da *string* da linha 1 deverá ser suficiente, sempre igual ou maior que a quantidade de itens do comando **DATA** (da rotina em linguagem de máquina).

Listagem 1

```
0 CLS
1 CLEAR: ST$="123456789012345678
901234567890"
2 I=VARPTR(ST$): 'I = LOCALIZACA
0 DO INDICE DA VAR. ST$
3 LSB=PEEK(I+1): 'LSB = BYTE MEN
OS SIGNIFICATIVO DA VAR. ST$
4 MSB=PEEK(I+2): 'MSB = BYTE MAI
S SIGNIFICATIVO DA VAR. ST$
5 L=LSB + 256 * MSB: 'L = LOCAL
IZACAO DA VAR. ST$
6 FOR Z=1 TO LEN(ST$)
7 PRINT@O,"CODIGO GRAFICO (128/2
55)": INPUT CG
8 POKE L+Z-1,CG
9 PRINT@470,ST$
10 NEXT
```

1 DE
RING
2 LS
LS+
3 IF
4 IF
6,LS
=L
5 RE
6 PO
10 Y
11 F
NEXT
SR(7
12 D
91,
13 D
46,2
15 C
20 R
25 I
30 F
\$(A\$
T
35 G
40 D

Para
listagem
cê deve
em ling



Contr
Mala
Conta
Tesou
Credia
Conta
Você enc
reveded

Listagem 2

```

1 DEFINT A-Z: CLEAR 100: ST$= ST
  RING$(30,0): V=VARPTR(ST$)
2 LS=PEEK(V+1): MS=PEEK(V+2): L=
  LS+256*MS
3 IF L>32767 L=L-65536
4 IF PEEK(16396)= 201 POKE 1652
6,LS: POKE 16527, MS ELSE DEFUSR
  =L
5 READ DT: IF DT=990 GOTO 15
6 POKE L,DT: L=L+1: GOTO 5
10 Y=USR (768+RND(10)): RETURN
11 FOR X=1 TO 15: Y=USR(768+X):
  NEXT: FOR X=15 TO 1 STEP -1: Y=U
  SR(768+X): NEXT: RETURN
12 DATA 205,127,10,62,1,24,0,237
  ,91,61,64,69,47,230,3
13 DATA 179,211,255,13,40,4,16,2
  ,46,24,242,37,32,241,201,990
15 CLS: 'PROGRAMA PRINCIPAL
20 READ A$: IF A$="FIM" GOTO 130
25 IF A$="1" GOSUB 11: GOTO 20
30 FOR A=1 TO LEN(A$): PRINT MID
  $(A$,A,1): GOSUB 10: NEXT: PRIN
  T
35 GOTO 20
40 DATA "OI, TUDO BEM?","
  "
50 DATA "CADA VEZ QUE VOCE ESCUT
  A ESTE BARULHINHO, ESTA' UTILIZA
  NDO A","LINGUAGEM DE MAQUINA ATR
  AVES DA FUNCAO 'USR' PELO COMAND
  O EM"
60 DATA "BASIC 'GOSUB' CONTIDO N
  A LINHA 30","
  "
70 DATA " EXPERIMENTE RETI
  RAR AS LINHAS - 6, 8 E 10 DO PRO
  GRAMA","'FORMULA 1' (MICRO SIST
  EMAS NR. 16, PAG.50) E ENTRE COM
  AS"
80 DATA "LINHAS DE 1 A 6 DESTA P
  ROGRAMA.", " ( NAO ESQUECER DE
  ENTRAR COM '990' NA LINHA 13 )",
  90 DATA "O SEU MICRO PROVIDENCIA
  RA TUDO PARA VOCE. E ASSIM, NAO
  SERA","NECESSARIO DESLIGA-LO PAR
  A RESPONDER PERGUNTAS COMO:",
  100 DATA "1"," 'READY?'
  "
110 DATA "1"," 'PROTEGE
  R"
120 DATA "FIM"
130 GOTO 130
  
```

Para terminar, rode o programa da listagem 2 e veja, na prática, como você deve agir para carregar suas rotinas em linguagem de máquina.

José Ricardo Flores Rodrigues é formado em Administração de Empresas e Ciências Contábeis pela Faculdade Cândido Mendes, RJ.

CPM

ONDE VOCÊ ENCONTRA
A SOLUÇÃO!

DESENVOLVIMENTO DE
SISTEMAS ESPECÍFICOS
PARA EMPRESAS

OFERTA ESPECIAL DO MÊS:
☐ CP-300 ☐ TK-2000

CPM Central Panamericana
de Microinformática
Pça Clóvis Beviláqua, 121 - 2º and. - SP
Tels (011) 32-7752 e 34-3057

**LANÇAMENTO
NACIONAL**

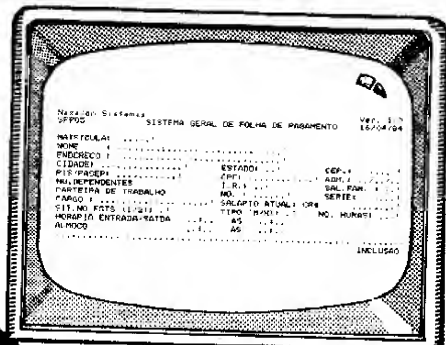
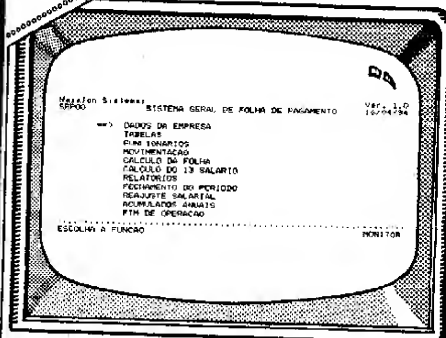
FOLHA DE PAGAMENTO.

A Nasajon Sistemas, uma empresa especializada no desenvolvimento de programas, está lançando no mercado, em caráter exclusivo, um completo sistema de Folha de Pagamento para microcomputadores dos tipos DGT 1000, CP 500, TRS 80 e outros. Relatórios emitidos: Relação de Empregados, Quadro de Horários, Folha de Pagamento, Resumo da Folha, Relação de FGTS, Guia de IAPAS, Relação de I.R., Guia de IAPAS, Relação p/Banco, Recibo de Pagto., etc.

Preço
Especial de Lançamento:
40 ORTN'S
Incluindo: diskete, manual completo,
tabelas e planilhas, assistência técnica
total e garantia de 1 ano.

nasajon
sistemas

Av. Rio Branco, 45 grupo 1311 - CEP 20090
Tel.: (021) 263-1241 - Rio de Janeiro - RJ



PROGRAMA	PREÇO EM ORTN'S
Controle de Estoque	20
Mala Direta c/ Ed. Texto	25
Contas a pagar/receber	15
Tesouraria (c/saldo bancário)	15
Credenciário	30
Contabilidade	30

Você encontra esses e outros programas em nossos
revendedores credenciados.

EQUIPAMENTOS

● Vendo APPLE II PLUS, language Card 64K, Drive, US\$ 5000 em Prog, preço 310 ORTN - Josias (021) 551-8350 - RJ.

● Vendo disquetes Verbatim-Datalife 5 1/4, tenho várias caixas com 10 cada. Tratar com Jorge Luiz Mara, Rua Timóteo da Costa, 444/804, tel.: (021) 274-5329, RJ.

● Vendo micro de bolso Sharp PC-1500, linguagem BASIC, interface para 2 cassetes e plotter com 4 cores e 9 tipos de caracteres. Na embalagem e com manuais. Tel.: (011) 240-4140, São Paulo.

● Troco TK 85 com 16K, joystick, gravador National, Mobylite, 11 fitas com jogos, 2 livros com aplicativos e jogos de vídeo (TV P&B) por CP500, ou por Color 64 excluindo vídeo. Tratar com Carlos a partir das 7 hs em dias úteis, tel.: (011) 421-4610 ou 421-4613. SP.

SOFTWARE

● Vendo ou troco Programas para computadores compatíveis com o TRS-80 Modelos I e III, e computadores de lógica Sinclair. Escreva para Carlos Seleri, Av. Sernambetiba, 3.600 B1 3/103, Rio de Janeiro, RJ, Cep 22.600.

● Troco/compro ou vendo programas para TK; aos interessados escrever para: Sylvio Padilha Jr., Rua 41-C, 139/41, Volta Redonda, RJ, Cep 27.180.

● Vendo fita com programa inédito do jogo "General" (poquer com dados), para micros Sinclair com 16K. Mandar cheque nominal de Cr\$ 7 mil para Silvio Rauth, Cx. Postal 8546, Cep 80.000, Curitiba, PR, tel.: 254-2881.

TROCO financeiro ofereço classificados VENDO alugo compro

DIVERSOS

● Traduzo programas das TI para BASIC do CP-500, CP-200 e compatíveis. Traduzo, também do BASIC, para a linguagem das TI desde que não exceda a memória de programação. Tratar com Hélio da Silva Araújo, Rua Des. Souto Maior, 244, Centro, João Pessoa, Paraíba, Cep 58000, ou com Sr. Neilton, tel.: (083) 221-9524 - horário comercial. Desenvolvemos, também, soft nas áreas comercial e de engenharia.

● Rádio amador: vendo interface para RTTY para os micros tipo Sinclair (TK 82/83/85 e CP-200). Informações com PY2-EMI, Renato Strauss, Rua Cardoso de Almeida 654/32, São Paulo, Cep 05013.

● Executamos alta resolução gráfica na linha Sinclair: TK82-C/83/85, CP-200, ZX81, Timex Sinclair 1000, etc. Preço de lançamento Cr\$ 35 mil. Acompanha soft explicativo. Paulo Roberto, tel.: 352-2710, após as 14 horas, RJ.

● Consultas sobre BASIC ou Assembler do Z80, Cx. Postal 57.041 Rio de Janeiro/RJ.

CLUBES

● Brasil Micro Clube, o maior clube do Brasil para usuários de todos os micros. Grandes vantagens e descontos para nossos associados. Associe-se gratuitamente através da Cx. Postal 40088, Rio de Janeiro, RJ.

● Gostaria de entrar em contato com usuários dos micros TK NE ou CP, para fundarmos o Micro Clube Natal, para troca de idéias, programas e informações. Tarcísio Torres de Sousa, Rua Joaquim Manoel, 801, Petrópolis, tel.: 222-4891, Cep 59000, Natal/RN.

● Gostaria de entrar em contato com possuidores de TK, e compatíveis, para troca de idéias, informações e programas, (aplicativos, jogos). Robson D. Klein: Rua Vereador Adão Rodrigues de Oliveira, 524, tel.: (0512) 934-285, Cep 93.300, Novo Hamburgo - RS.

● Gostaria de contatar com possuidores de TRS 80-Color para troca de idéias, programas, manuais e dicas. Dirija-se a José Alberto Cadeais, QI 23 conj. 17 Casa 01, Lago Sul, CEP: 71600, Brasília, DF. Tel.: (061) 571-1429.

● Quero entrar em contato com possuidores do TK-2000 Color para troca de programas e dicas. Tratar com Jorge pelo tel.: (071) 231-1227, ou pelo endereço: Alameda das Cajazeiras, 43, Caminho das Árvores, Cep 40000, Salvador, Bahia.

CURSOS

● O SENAC está promovendo, de 11 a 19 de junho, o curso Implantação do Microcomputador: novo enfoque tecnológico e administrativo. Esse curso é destinado a empresários, executivos, analistas de O & M e analistas de sistemas. Informações na Rua Dr. Vila Nova, 228, 2º andar, fone: (011) 256-5522, São Paulo, SP.

● A PRO-INFORMÁTICA, SISTEMAS CONSULTORIA E TREINAMENTO oferece regularmente cursos de COBOL, BASIC, BASIC AVANÇADO e ASSEMBLY com turmas reduzidas. Além das aulas práticas, os alunos dispõem dos computadores com assistência de monitores em horários extra-aula. RUA JOSAFÁ BELO 100 - 337-8792 - CIDADE JARDIM - BELO HORIZONTE.

● O CBI - CENTRO BRASILEIRO DE INFORMÁTICA promove para os meses de ABRIL/MAIO os cursos de BASIC, BASIC AVANÇADO, MUNPS, COBOL, FORTRAN, ASSEMBLER, MANUTENÇÃO DGT 100/1000. Turmas reduzidas, certificado de conclusão e aulas práticas. Informações na Av. Passos nº 115 Sala 215 ou pelo telefone 233-1123.

● A MIKRO INFORMÁTICA continua oferecendo com sucesso os seguintes cursos: Informática p/jovens, Operação e Programação de microcomputadores, Linguagem Basic e Basic Avançado. Inscrições e informações à Av. Af. Pena 952/522 Tel. - 222-3035 BH/MG.

● Em Niterói, a MICROWARE Informática e Eletrônica promove regularmente cursos de Introdução aos Microcomputadores, de linguagem BASIC e Assembler, com aulas práticas em equipamentos DGT-1000. As turmas são limitadas em 12 alunos. O material didático é gratuito e é fornecido certificado. Inscrições e informações na Rua Moreira César, 229 - sala 1713, Shopping Icaraí, Niterói, tel.: (021) 710-2780, RJ.

● A FAAP oferece regularmente cursos de BASIC Básico (27 horas) e Avançado (45 horas) nos TK 85, CP-200/300/500 e Júnior Itautec. Horários pela manhã, tarde e noite. Maiores informações pelo tel.: (011) 66-2147 ou 826-4233, ramal 59, São Paulo, SP.

A partir do número 33,
edição de junho

QUEM MANDA NESTA PÁGINA SOU EU!

Apoiado! Equipamentos, Software, Cursos, Clubes e Diversos: você é quem decide o que, quando e como anunciar nos Classificados MS. Quanto você terá que pagar? Isso também é decisão sua. Preste atenção:

- cada linha de texto (30 toques, incluindo os espaços em branco) custa Cr\$ 2.000,00;
- linhas incompletas serão cobradas como inteiras;

● o próprio anunciante deve checar o valor de seu anúncio com o número de linhas que ele contém;

● o anúncio deve vir acompanhado de um cheque nominal à ATI Editora Ltda;

Os textos devem ser datilografados ou escritos em letra de fôrma, obedecendo as 30 batidas por linha. Veja um exemplo:

V	e	n	d	o		D	G	T	-	1	0	0		c	o	m		3	2		K		R	A	M	,		v	i
d	e	o		e		g	r	a	v	a	d	o	r		c	a	s	s	e	t	e	.		T	r	a	t	a	r
c	o	m		M	a	r	c	o	s	,		t	e	l	:	(0	2	1)	2	6	7	-	0	3	3	2	.

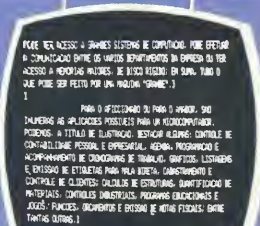
Maiores informações pelos telefones: (021) 262-5259 - RJ ou (011) 853-7758 - SP.

**Micro
Sistemas**

A geração definitiva é sempre a próxima.



DEFENSE COMMAND



SYSWORD



PENETRATOR



SYSCALC



SCARFMAN



DANCING DEMON

Você só descobre o quanto precisa de um Micro-Computador **JR** da **Sysdata** depois que o conhece de perto.

Você vai ter certeza de que fez um ótimo negócio ao adquiri-lo assim que o colocar na sua empresa ou na sua casa.

O JR da Sysdata é rápido, é versátil, é compacto.

APLICAÇÕES:

Contabilidade, controle de contas a pagar, controle de contas a receber, folha de pagamento, controle de estoque, controle de clientes, relatório de clientes, mala direta, cálculos de orçamentos financeiros, controle de processos industriais, cálculos de engenharia, cálculos de estatísticas, funções matemáticas, funções lógicas em cadeia de caracteres (STRINGS), gráficos, jogos animados, programas educacionais.

O JR PERMITE AINDA:

O acesso a grandes sistemas de computação, a comunicação entre os departamentos de Empresa, efetuar programas específicos para cada Empresa.

E, como se não bastasse, ele é o Micro-Computador **de menor preço do mercado.**

Com todas as qualidades que tem, o JR da Sysdata nem precisava ser tão econômico. Mas é.

Afinal, ele é o mais completo Micro-Computador de sua geração.

Inclusive no preco.

Você pode testar estas e outras qualidades do JR em qualquer dos nossos revendedores.

JR Sysdata
Microcomputador pessoal

Microcomputador pessoal

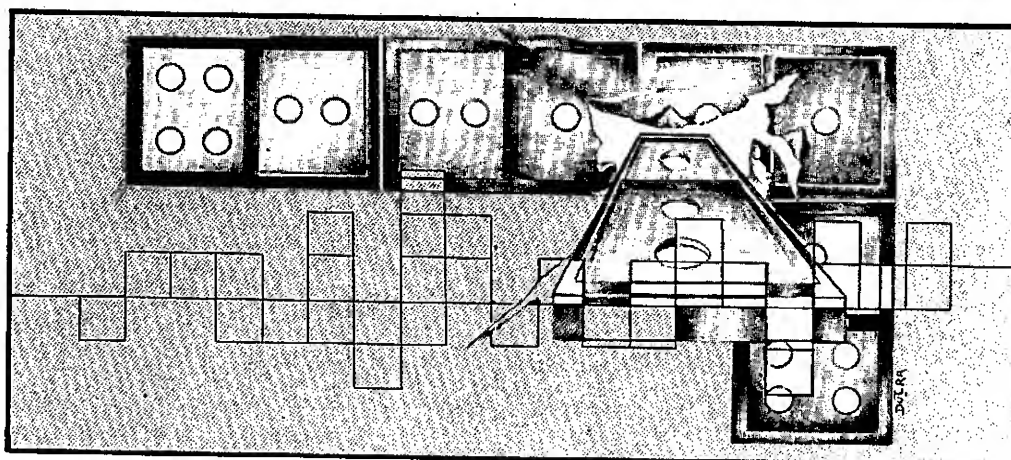


Sysdata
eletrônica ltda

AV. PACAEMBÚ, 788
CEP 01155 - TEL: 67.5900

REVENDEDORES: SÃO PAULO/Capital - Ad Data 864.8200; ADP System 227.6100; Búcker 881.7995; Cinótica 36.6961; Computemarketing 212.9004; Compute 852.8533; Computelrad 231.3277; Foto Léo 35.7131; Fotográfica 853.0448; Guedes 289.9051; Horst 203.5597; Interface 852.5603; Lema 210.6929; Microrei 881.0022; Miprotec 269.4941; Nova Geração 814.3663; O.P.A. 35.8685; Plantasia 275.0181; Plantel 543.9653; Sacco 814.0598; Servimec 222.1511; Sistencar 282.6609; S.O.S. 66.7656; Runner's 469.0887; **Campinas** - Computer House 852.5855; Computique 32.6322; Microtok 32.4445; **Rio Claro** - Coml. Micro Cosmos 34.5801; **Ribeirão Preto** - Compuspy 635.1195 - **Araras** - Copec 41.3779; **Taubaté** - Enscion 33.2252; **Mogi Guaçu** - Guacamapa 261.0236; **Bragança Paulista** - Infordata 543.5198; **Bauru**; **Marília** - Sipro 33.4109; **Catanduva** - Teledata 22.8119; **RIO DE JANEIRO/Capital** - Clap 228.0734; Computique 267.1093; G D M Informática 284.8744; JR de Góes 246.4180; Kristian 391.3165; Suprimento 274.8845; **Petrópolis** - Foto Oti 42.1391; **MINAS GERAIS/Belo Horizonte** - Compucity 226.6336; Computec 225.2617; Kemirton 25.0644; Minas Digital 337.7946; **Poços de Caldas** - Computique 721.5810; **RIO GRANDE DO SUL/Porto Alegre** - Advancing 26.1194; Alptec 24.0465; Digital 24.1411; Microsis 22.9782; **Pelotas** - Sistemátika 22.3810; **Novo Hamburgo** - Micromega 93.4721; **PARANÁ/Curitiba** - Computique 243.1731; Micro System 232.3533; Morgen 322.0593; **Ponta Grossa** - Grupo Data Memory 24.6191; **Londrina** - Shop Computer 23.9674; **GOIÁS/Goiânia** - Casa do Microcomputador 223.1165; Grupom 225.8226; **SANTA CATARINA/Florianópolis** - Castro 22.6933; Infotec 23.4777; **BRASILIA/Distrito Federal** - Compushow 273.2128; Digitec 225.5434; **MATO GROSSO DO SUL/Campo Grande** - DRL 382.6487; Video 321.4220; **CEARÁ/Fortaleza** - Siscompy 244.4691; **PARAIBA/João Pessoa** - Medusa 221.6743; **PEPAMBUCO/Recife** - Eloquia 241.1388.

Dominó é um jogo que anda meio esquecido por aí. Mas o seu micro TRS-80 o redescobriu, gostou e quer desafiá-lo para uma partida (com direito a musiquinha e tudo)



Domine o micro no dominó

Everton Pereira

Quanto tempo faz que você não disputa uma partida de dominó? Se é por falta de adversário, eis aqui a sua oportunidade de enfrentar um adversário inédito: seu micro! E você terá ainda uma grande vantagem: ele não é nada temperamental. Se você o estiver vencendo uma partida, não correrá o risco de seu adversário esbarrar na mesa de jogo e, *sem querer*, espalhar as pedras.

Este programa pode ser rodado em qualquer micro compatível com o TRS-80 modelo III. Quanto às instruções de jogo não há muito o que dizer, uma vez que, além de ser um jogo muito conhecido de todos, ele é auto-explicativo, ou seja, a cada jogada o micro lhe dirá o que fazer para prosseguir.

Fica a sugestão para que os leitores façam modificações no programa. Por exemplo: vocês podem alterar a tonalidade e/ou ritmo do som ou criar diferentes níveis de dificuldade para o jogo. Eis aqui duas dicas:

1) o endereço 30007 (vide linha 5010) determina a tonalidade do som que acompanha a impressão da palavra DOMICRO, podendo variar de 0 a 225 (do agudo para o grave);

2) caso o leitor ache o programa muito extenso poderá reduzi-lo suprimindo as linhas 5010 a 5030 e 9510 a 9570 (sub-

rotina de impressão da palavra DOMICRO), criando então a linha 4050 GO-SUB 8400: NEXT: IJ= 52.

Formado em Ciências Econômicas, Everton Pereira é funcionário do Banco do Brasil, agência de Patrocínio Paulista, SP. Aprendeu a linguagem BASIC num NE-Z8000 e atualmente é usuário de um CP-300.

Domicro

```
0 POKE16526,48:POKE16527,117:RANDOM
10 CLEAR1000:DEFINT I:DEFSTR S:DIMSP(36):DIM IE(4)
20 FORIB=30 TO 31:FOR IA=1 TO 5:READ IC:SP(IB)=SP(IB)+CHR$(IC):N
EXT: NEXT:SP(32)=STRING$(5,131):FOR IB=33 TO 36:FOR IA=1 TO 3:RE
AD IC:SP(IB)=SP(IB)+CHR$(IC):NEXT: NEXT
35 SE=CHR$(151)+CHR$(131)+CHR$(171):SF=CHR$(149)+CHR$(32)+CHR$(1
70):SG=CHR$(157)+CHR$(140)+CHR$(174):SH=CHR$(181)+CHR$(176)+CHR$
(186)
70 FORIA=30000 TO 30015:READ IB:POKE IA,IB:NEXT:IB=30:IW=5:IK=0:
CLS:GOSUB4010:IT=0:IP=IT
180 IFIY<15 THEN PRINT@654,"voce que comeca, pois tem o duble 6/
6":IY=2:ELSE PRINT@652,"SOU EU QUE COMECO POIS TENHO O DUBLE 6/
6":IY=1
185 IB=35:IC=30:IH=5:IW=IH:IK=0:IE(1)=6:IE(3)=IE(1):FOR IA=1 TO
2000:NEXT
215 ON IY GOSUB 1050,2000:IFIL>1 AND IT=IP THEN IT=14:IP=14 ELSE
IF IL>1 THEN IP=14*-1*(IP>IT):IT=14*-1*(IT)=IP)
220 IF IT=14 AND IT>IP THEN PRINT@645,STRING$(23,"*")"EU GANHEI"
STRING$(23,"*"), ELSE IF IP=14 AND IP>IT THEN PRINT@645,STRING$(22
,"*")"VOCE GANHOU"STRING$(22,"*"),ELSE IF IT=14 AND IP=IT THENPR
INT@645,STRING$(22,"*")"EMPATAMOS"STRING$(22,"*"),
230 PRINT@773,STRING$(55," "),IF IT=14 OR IP=14 THEN 600 ELSE G
```

```

010 215
600 IL=USR(0):FOR IA=1 TO 50:NEXT:PRINT@786,STRING$(29,32):FOR
IA=1 TO 50:NEXT:PRINT@786,"VOCE QUER JOGAR NOVAMENTE?S/N":SP(29
)=INKEYS:IF SP(29)("&S" AND SP(29)("&N" THEN 600 ELSE IF SP(29)=
"8" THEN RUN
710 PRINT@783,"OBRIGADO POR TER JOGADO COMIGO. . .":GOTO710
1050 POKE30007,30:IY=2:PRINT@645,STRING$(55,""),@773,STRING$(54
,""),@659,"AGUARDE. . . ESTOU PENSANDO":IL=IL+1:FOR IV=15 TO 2
8:IF SP(IV)("&") THEN GOTO 3200
1200 NEXT:RETURN
2000 POKE 30007,100:IY=1:PRINT@646,"SUA VEZ, DIGITE A LETRA CORR
ELATA A PEDRA A SER JOGA-":@774,"DA. SE VOCE NAO TIVER P/JOGAR,
DIGITE APENAS ENTER":
2020 SP(29)=INKEYS:IF SP(29)="" THEN 2020 ELSE IV=ASC(SP(29))-64
:IL=IL+1:IF IV=14 OR IV=1 THEN RETURN
3200 SP(29)="E":IE(2)=VAL(MID$(SP(IV),1,1)):IE(4)=VAL(MID$(SP(IV
),2,1))
3300 IFSP(IV)="" OR (IE(1)()IE(2) AND IE(3)()IE(4) AND IE(1)()I
E(4) AND IE(2)()IE(3)) THEN IF IY=2 THEN 1200 ELSE IF IY=1 THENR
ETURN
3400 IF IY=2THEN IT=IT+1:PRINT@258+IT*4," ", ELSE IP=IP+1:PRINT
@450+4*IV," ",@513+4*IV," "
3700 IF IE(1)()IE(3) AND(IE(2)=IE(1) OR IE(3)=IE(4)) THEN SP(IV)
=MID$(SP(IV),2,1)+MID$(SP(IV),1,1):IE(2)=VAL(MID$(SP(IV),1,1)):I
E(4)=VAL(MID$(SP(IV),2,1))
3750 IF IY=1 ANDIE(1)=IE(4) AND IE(3)=IE(2) AND IE(1)()IE(3) THE
N PRINT@774,STRING$(30,32):FOR IJ=1 TO 50:NEXT:PRINT@774,"QUAL
LADO VOCE QUER JOGAR? E/D"STRING$(24,""),SP(29)=INKEYS:IF SP(2
9)("&D" OR SP(29)("&E" THENFOR IJ=1 TO 50:NEXT:IJ=USR(0):GOTO 3750
3800 IF IE(1)=IE(4) AND SP(29)=""E"THEN GOSUB 8400:IE(1)=IE(2) EL
SE GOSUB 8000:IE(3)=IE(4)
3900 SP(IV)=""":RETURN
4010 IP=15:IT=1:FOR IA=1 TO 28
4020 IC=RND(2)
4030 IFIC=1 AND IT<15 THEN READ SP(IT):IT=IT+1 ELSE IF IP<29 THE
N READ SP(IP):IP=IP+1 ELSE GOTO 4020
4040 IF SP(IT-1)=""66" THEN IY=IT-1 ELSE IF SP(IP-1)=""66" THEN IY
=IP-1
5010 GOSUB 8400:NEXT:IA=327:IB=IA:IH=0:POKE30007,30
5020 FOR IC=1 TO 4:FOR IH=0 TO 6+3*(IH=3):READ IH:IF IH<32 THEN
PRINT@IA+IW,CHR$(IH):IJ=USR(0)
5030 NEXT:IA=IA+64:NEXT:IH=IH+1:IB=IB+8+3*(IH=4):IA=IB:IF IH<7 T
HEN 5020 ELSE IJ=52
7017 FORIA=0 TO 5:GOSUB 9300:PRINT@966,"":NEXT:GOSUB 9300:PRINT@
988,"MINHAS PEDRAS":GOSUB 9300:PRINT@966,"":GOSUB 9300:PRINT@965
,STRING$(55,140):GOSUB 9300:PRINT@989,"SUAS PEDRAS":FOR IA=1 TO
2:GOSUB 9300:PRINT@966,"":NEXT:GOSUB 9300:PRINT@965,STRING$(55,"
")
7040 FOR IA=1 TO 5:GOSUB 9300:PRINT@966,"":NEXT:POKE30007,50:FOR
IA=4 TO 56 STEP 4:IJ=USR(0):PRINT@258+IA,"",@450+IA,CHR$(64+IA
/4),@513+IA,MID$(SP(IA/4),1,1)"/"MID$(SP(IA/4),2,1):NEXT:RETURN
8000 IL=0:IF IC=-10 THEN IC=0:I0=0 ELSE IF IC=70 THEN IC=60:I0=8
32
8100 IC=IC+IH:IF IC=-10 OR IC=70 THEN IH=IH*(-1)
8200 IK=IC:IZ=0:IM=I0:GOTO8700
8400 IL=0:IF IB=-10 THEN IB=0:I0=832 ELSE IF IB=70 THEN IB=60:I0
=0
8500 IB=IB-IW:IF IB=-10 OR IB=70 THEN IW=IW*(-1)
8600 IK=IB:IZ=0:IM=I0
8700 IF IK<55 THEN IZ=(IK-60)/5 ELSE IZ=ABS(IK/5)
8750 IFIK<5 AND IK<60 THEN 8800 ELSE IF IM<27 THEN I0=2+320*IZ
-58*(IK/50):GOTO8900 ELSE I0=642-320*IZ-58*(IK/50):GOTO 8900
8800 IJ=USR(0):PRINT@IK+IM,SP(30):@IK+IM+64,SP(31):@IK+IM+128,SP
(32):@IK+IM+65-2*(IM=832),MID$(SP(IV),1,1):@IK+IM+67+2*(IM=832),
MID$(SP(IV),2,1):RETURN
8900 IJ=USR(0):PRINT@IG,SP(33):@IG+64,SP(34):@IG+128,SP(35):@IG+
192,SP(36):@IG+256,SP(36):@IG+193+128*(IK/55),MID$(SP(IV),1,1):@
IG+65-128*(IK/55),MID$(SP(IV),2,1):RETURN
9300 POKE 30007,IJ:IJ=IJ-3:IZ=USR(0):RETURN
9480 DATA 156,140,156,140,172,149,32,169,32,170,151,131,171,149,
32,170,157,140,174,181,176,186
9490 DATA 211,0,214,2,211,254,6,30,16,254,43,120,181,32,239,201
9500 DATA"00","11","22","33","44","55","66","01","02","03","04",
"05","06","12","13","14","15","16","23","24","25","26","34","35",
"36","45","46","56"
9510 DATA 191,159,143,143,143,189,144,191,149,128,128,128,191,14
9,191,149,128,128,128,191,149,191,189,188,188,188,159,129
9520 DATA184,159,143,143,143,189,144,191,149,128,128,128,191,149
,191,149,128,128,128,191,149,139,189,188,188,188,159,129
9530 DATA 191,189,144,128,184,191,149,191,149,139,158,129,191,14
9,191,149,128,128,191,149,191,149,128,128,128,191,149
9540 DATA139,191,159,129,128,191,149,128,128,191,149,128,184,191
,189,144
9550 DATA 184,159,143,143,143,143,133,191,149,128,128,128,128,12
8,191,149,128,128,128,128,139,189,188,188,188,188,148
9560 DATA 184,159,143,143,143,189,144,191,181,176,176,176,191,13
3,191,151,131,191,151,129,128,191,149,128,138,175,176,176
9570 DATA 184,159,143,143,143,189,144,191,149,128,128,128,191,14
9,191,149,128,128,128,191,149,139,189,188,188,188,159,129

```



TECNOLOGIA EFICIÊNCIA SERIEDADE

a serviço de sua
empresa

Software e Suprimentos
para Computador
Mariz e Barros 711
Tijuca - RJ
Fone:
284-3490 — 284-3586

"FILIADA À ASSESPRO."

POSSUIMOS PROGRAMAS DE ALTO DESEMPENHO PARA MICROCOMPUTADORES.

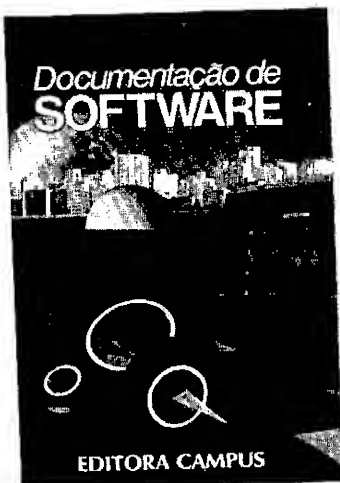
ÁREA ADMINISTRATIVA

- Contabilidade
 - Ativo Imobilizado
 - Relatório Gerencial
 - Mala Direta
 - Contas a Receber
 - Contas a Pagar
 - Controle Bancário
 - Controle de Estoques
 - Folha de Pagamento
 - Faturamento
 - Editor de Texto
- ### ÁREA EDUCATIVA
- Learning English

INTERFAC
SOFT

Avenida do Contorno 6656
L 19 Belo Horizonte 30000
(MG) Tel: (031) 223 4133

Estamos credenciando revendedores
e representantes em todo Brasil.



LOMAX, J. D., *Documentação de Software*, Editora Campus.

Este livro se destina a qualquer leitor interessado na fabricação (desenvolvimento, fornecimento e manutenção) ou no uso de produtos de software. Os primeiros, ele auxilia na complexa tarefa de elab-

borar instruções que possibilitem uma utilização ampla e eficaz de seus produtos; aos usuários, ele mostra a importância dessa documentação como parte essencial do software, apontando-lhes os aspectos que deverão levar em conta durante a seleção do software, ou que precisarão discutir com seu futuro fornecedor.

Os três capítulos iniciais analisam separadamente cada um dos tipos de manuais e o último deles ensina a elaborar textos que propiciem uma comunicação simples e objetiva entre clientes e fabricantes.

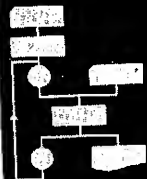
ALVES, A., *Programação*, Editora Atlas.

Este livro é pioneiro no tratamento da técnica de programação. Não é um texto sobre linguagem de programação ou análise de sistemas e sim sobre os problemas operacionais vinculados à codificação, aos testes, à depuração de erros e à documentação.

O autor dividiu o livro em cinco capítulos. O primeiro trata dos Projetos, onde se ensina a elaboração de fluxogramas dentro das normas da ANSI/70. O segundo

ALOISIO PINTO ALVES

Programação



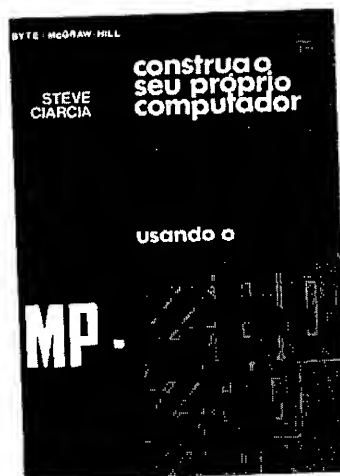
- CODIFICAÇÃO
- TESTES
- DEPURAÇÃO DE ERROS
- DOCUMENTAÇÃO

ATLAS

capítulo é dedicado à Codificação, contendo tópicos relacionados com a clareza de um programa, e no terceiro vêm os Testes, apresentando sugestões para localização de erros. No capítulo 4 é apresentada a Depuração de Erros e no capítulo 5 a Documentação, com informações sobre a finalidade e leitura do programa.

etapas do planejamento estratégico e tático.

O quinto capítulo ressalta as providências a serem tomadas na parte organizacional e na técnica e, finalmente, o sexto trata do controle do planejamento.



CIARCIA, S., *Construa o seu próprio computador usando o MP-280*, Editora McGraw-Hill.

Este guia prático mostra como construir um computador baseado no famoso microprocessador Zilog Z80. A descrição enfoca um microcomputador básico em placa única contendo 2K de sistema operacional, portas serial e paralela, display hexadecimal e armazenagem de massa em fita cassete. Cada subsistema do computador é completamente explicado e calçado em informações provadas e testadas de forma a que o leitor possa facilmente modificar o sistema (adicionando, por exemplo, um terminal de vídeo) a fim de satisfazer suas necessidades pessoais.

BORGES, J. A., *BASIC - Aplicações Comerciais*, Editora LTC.

A abordagem geral do livro se baseia em exemplos voltados para aplicações do tipo gerencial. Partindo de programas completos, que são minuciosamente analisados, o leitor é guiado ao entendimento do computador e da atividade de programação. Todos os detalhes da linguagem BASIC são explorados, inclusive os aspectos que não são padronizados nas diversas versões para diferentes computadores. Especialmente, o capítulo sobre arquivos fornece a base para desenvolvimento das aplicações de uso na administração em geral. Embora ênfase seja dada no BASIC para micros, este texto poderá ser utilizado por programadores que utilizem o BASIC em computadores de maior porte.

CARLOS ALBERTO C. ABREU

77 programas para linha apple

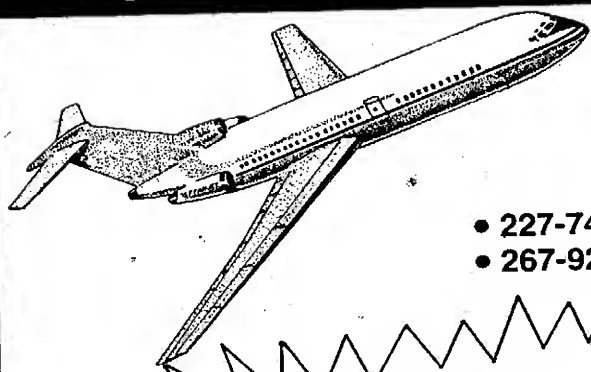


ABREU, C. A., *77 Programas para linha Apple*, Edição Micro-Kit.

Este livro serve como texto didático para aqueles que estão começando a programar em BASIC e também como material de apoio para os estudantes de 1º e 2º graus. Com ele, além de passar horas divertindo-se com seu computador, você poderá observar, com o auxílio dos comentários, como são construídos os programas e a sua lógica.

Além de jogos, o livro traz vários programas educacionais como: Decomposição de fatores primos; Como calcular a área de um polígono; Método de Simpson; Operações de vetores; Divisão de dois números; Plotagem de curva e outros, os quais lhe darão a oportunidade de estudar Matemática através do computador.

SEJA PILOTO OU MECÂNICO DE AVIÕES OU HELICÓPTEROS



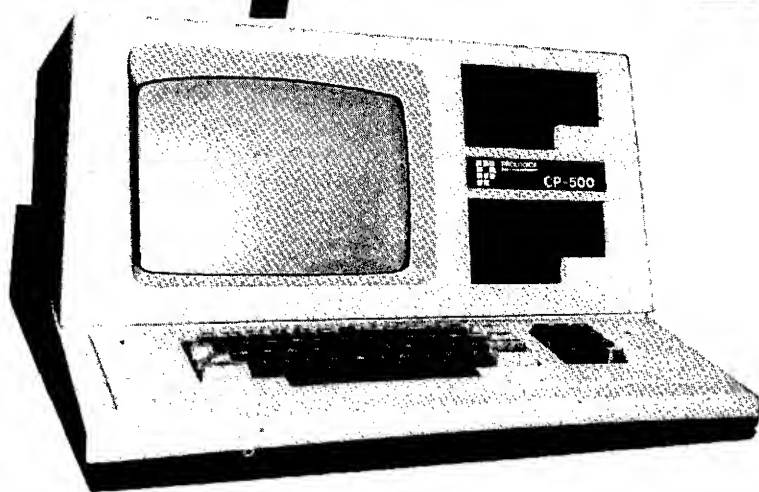
• 227-7417
• 267-9261

NOVAS TURMAS
MATRICULE JÁ!

RIO DE JANEIRO

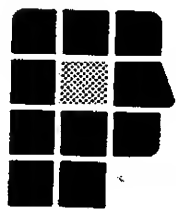
EMPAER^R

Comprar o CP 500 na Clappy não é só comprar um CP 500



Você pode até achar que o preço e as condições que a Clappy oferece são as melhores do mercado. O que é verdade, fácil de você comprovar. Mas o que a Clappy deseja é fazer muito mais por você. Daí, ao comprar seu micro na Clappy, veja o que ela lhe oferece:

- Departamento de Consultoria e Apoio ao Comprador
- Curso de Programação e Operação
- Treinamento
- Implantação, fornecimento e instalação de sistemas
- Grande variedade em Softwares, periféricos e suprimentos.



PROLOGICA
microcomputadores

Clappy

Venha à nossa loja ou solicite
a visita de um representante.

COPACABANA:

Rua Pompeu Loureiro, 99
Tel.: (021) 236-7175 - 257-4398
Aberta diariamente das 10 às 20 horas e aos sábados das 10 às 15 horas. Estacionamento próprio.

CP 500, nas versões cassete, 1 ou 2 discos.
Placa CP/M. Sistema 700 com super file e
impressora de 200cps, disco Winchester de
5 Mb. Sistema 600, PROCALC (o visicalc do
CP 500), Banco de Dados básico e avançado,
processador de textos básico e super,
compiladores BASIC, COBOL, FORTRAN e
PASCAL. Além de uma infinidade de jogos.

APLICATIVOS COMERCIAIS:

Contabilidade, Controle de estoque, Folha
de pagamento, Contas a pagar e receber.

APLICATIVOS DE APOIO À DECISÃO:

Planilha financeira, Processamento de
textos, Mala Direta, Cadastro de clientes,
Controle financeiro.

A Clappy desenvolve soluções específicas
para qualquer problema que você tenha.

A CLAPPY TEM A MELHOR SOLUÇÃO PARA SEUS PROBLEMAS.

CENTRO:

Av. Rio Branco, 12 - Loja e sobreloja
Tel.: (021) 253-3395
Rua Sete de Setembro, 88 - Loja Q (Galeria)
Tel.: (021) 222-5517 - 222-5721

Entregamos em todo Brasil pelo reembolso Varig.

Linha Apple

Disco voador sonoro

Lendo o útil e interessante artigo *Rotinas de som e animação gráfica*, de Rudolf Horner Junior, na Seção Sidra de MS nº 27, chamou-me a atenção o som do disco voador. Resolvi fazer uma pequena alteração no programa (espero não ferir os sentimentos do autor do artigo) e juntar a imagem ao som.

Depois de rodar o programa, é interessante afastar-se cerca de quatro metros da TV, pois o efeito fica ainda mais atraente e sugestivo.

```
10 REM DISCO VOADOR * BY CARIBA * DEZ 1983
20 HOME
30 DATA 160,1,162,0,138,24,233,1,208,252,1
41,48,192,232,224,255,208,242,236,208,237,
96
40 FOR A=768 TO 789 : READ B : POKE A,B :
NEXT
50 GR : COLOR = RND(16)*16
60 HLIN 17,21 AT 17
70 HLIN 15,23 AT 18
80 HLIN 12,26 AT 19
90 HLIN 9,29 AT 20
100 HLIN 7,31 AT 21
110 HLIN 9,29 AT 22
120 HLIN 13,25 AT 23
130 HLIN 15,16 AT 24
140 HLIN 22,23 AT 24
150 FOR A=1 TO 255 STEP 3
160 POKE 774,105 : POKE 769,1 : POKE 783,A
170 POKE -16304,A
180 CALL 768
190 POKE -16303,0
200 FOR B=1 TO 30 : NEXT B,A
210 GOTO 50
```

Carlos Ribeiro de Barros-SP

Linha TRS-80

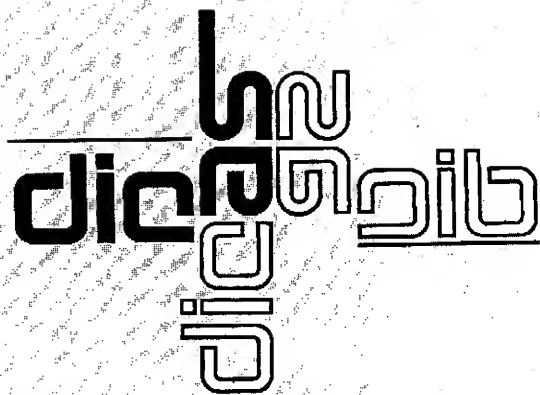
Bye bye INKEY\$

Sabendo os endereços e valores de cada tecla, fica muito fácil eliminar o INKEY\$ e permitir maior velocidade nos programas.

Observe que neste programa aparecem no vídeo os valores relacionados a cada tecla, e se apertarmos várias teclas simultaneamente, o resultado que surgirá na tela será a soma dos valores de cada tecla. Assim, para substituir o INKEY\$, basta usar IF PEEK (endereço) = (valor da tecla) THEN....

```
1 CLS : FOR K=0 TO 7 : X(K) = 1433
6+2*K : PRINT@128*K,X(K) : NEXT
T
2 FOR K = 0 TO 7 : PRINT@128*K+8
,PEEK(X(K)), : NEXT : GOTO2
3 REM ENDERECOS E VALORES DO TEC
LADO
4 REM ALVARO DE FILIPPO (031)
823-1948 IPATINGA MG
```

Álvaro de Filippo-MG



Se você tem pequenas rotinas e programas utilitários realmente úteis tomando poeira em seus disquetes ou fitas cassetes, antecipe-se aos piratas e trate de divulgá-los. Envie-os para a **REDAÇÃO DE MICRO SISTEMAS - SEÇÃO DICAS: Av. Presidente Wilson, 165/grupo 1210, Centro, Rio de Janeiro, RJ, CEP 20030. Não se esqueça de dizer para qual equipamento foram desenvolvidos. Desta forma, sua descoberta poderá ser útil para muitos e muitos, em vez de desmagnetizar-se com o tempo em suas fitas e disquetes...**

Linha TRS-80

Duplicador de linhas

Este utilitário gera uma linha idêntica a outra já existente no programa, pois não existe esta função no BASIC dos micros compatíveis com a linha TRS-80. Para operá-lo, basta seguir a sequência:

— digite o número da nova linha;

— utilize a tecla de tabulação (→) para mover o cursor, reservando um espaço idêntico ou superior ao

da linha que será duplicada; — digite o símbolo de comentário (') e, como lembrete, o número da linha que será duplicada;

— tecle RUN 65000 ou GOTO 65000 e responda às perguntas "DE" e "PARA" com os respectivos números das linhas;

— tecle < ENTER > e observe que você obteve uma cópia fiel de uma linha já existente.

```
65000 INPUT"DE";L1 : INPUT"PARA";L2 : L3=165
48
65001 L3=PEEK(L3+1)*256+PEEK(L3) : L3=L3+(L3
32767)*65536 : IF L3=0 THEN 65000 ELSE
E E=PEEK(L3+3)*256+PEEK(L3+2) : IF E<>
L1 AND E<>L2 THEN 65001
65002 IF E=L1 THEN E1=L3+4 ELSE E2=L3+4
65003 IF E1=0 OR E2=0 THEN 65001
65004 POKE(E2),PEEK(E1) : E1=E1+1 : E2=E2+1
: IF PEEK(E1)>0 AND PEEK(E2)>0 THEN 65
004 ELSE PRINT"OK" : EDIT.
65005 ' DUPLICADOR DE LINHAS * ALVARO DE FIL
IPPO * (031) 823-1948
```

Álvaro de Filippo-MG

Linha Apple

Formatando quantias monetárias

Esta dica transforma qualquer valor numérico (exceto em notação científica, tipo 1.234 E+3) positivo ou negativo para o formato XXX.XXX,XX, que é muito utilizado para representar quantias monetárias.

O programa deve ser usado como sub-rotina num programa principal, e para melhor compreensão vamos analisá-lo a partir da figura 1: vamos supor a variável VENDAS = 1.000.856 e LUCRO = VENDAS*0.8, ou seja, LUCRO = 800.6848. Para formatar os valores de VENDAS e LUCROS, usaremos as linhas 110 e 120 da figura 1, colocando numa variável VX o valor da variável a ser formatada. Ativamos a sub-rotina formatadora (GOSUB 4000), que nos devolve o valor formatado na variável Z\$ e salvamos o valor desta variável numa outra variável alfanumérica qualquer (no caso da figura 1, LUCRO\$ e VENDAS\$).

```
10
20 VENDAS = 1.000.856
30
:
100 LUCRO = VENDAS * 0.8
110 VX = VENDAS : GOSUB 4000 :
    VENDAS$ = Z$
120 VX = LUCRO : GOSUB 4000 :
    LUCRO$ = Z$
:
40000 Z$ = "": .....
40010
:
40190 RETURN
```

Figura 1

É importante evitar que as variáveis utilizadas na sub-rotina não sejam usadas no programa principal, pois poderá gerar perda de valores e criar confusão em outros resultados. Para evitar a repetição, fixe as variáveis usadas: VS, VOS, Z\$, VX, FLAG, CNT, MOD, L. Por fim, é interessante destacar que esta dica aceita até o valor máximo de 999.999.999,00.

Sub-rotina formatadora

```
40000 Z$ = "": VS = STR$ (VX) : CNT
    = 0
40010 IF SGN ( VAL (VS)) = - 1
    THEN FLAG = 1
40020 VS = STR$ ( ABS ( VAL (VS)
    ))
40030 FOR I = 1 TO LEN (VS)
40040 IF MID$ (VS,I,1) = "." THEN
    GOTO 40070
40050 NEXT I
40060 VOS = ".00": GOTO 40080
40070 VOS = "." + MID$ (VS,I + 1
    ,2)
40080 FOR X = 1 - 1 TO 1 STEP -
    1
40090 CNT = CNT + 1
40100 IF (CNT / 3 - INT (CNT /
```

Antonio Carlos Palmeira Salles-PB

Linha Sinclair

Zerando qualquer linha

A dica "Zerando REM", publicada na Seção Dicas de MS nº 27, zera apenas as linhas de comentário dos programas em BASIC. Com pequenas alterações, entretanto, podemos fazê-la zerar qualquer linha de programa. Para tal, digite a dica "Zerando REM" e faça as seguintes alterações:

```
9870 INPUT X
9960 IF PEEK (L+5) < > X
    THEN RETURN
```

A linha 9870 recebe o código da instrução cujas linhas serão zeradas. Se, por exemplo, X= 245, então todas as linhas PRINT receberão o número 0 e não poderão ser mais deletadas. Com relação às outras instruções (IF, LET, INPUT etc.), consulte o manual do seu equipamento.

Rodrigo Loureiro Pinto-RJ

Linha TRS-80

Otimização de caracteres duplos

Se você quiser retornar da condição de caracteres de tamanho duplo (32 cpl) sem ter que usar CLS e com isso apagar toda a tela, digite em BASIC:

```
POKE 16912,40 : OUT 236,0 < ENTER > (Mod III)
POKE 16445,0 : OUT 255,0 < ENTER > (Mod I)
```

A única coisa diferente que acontecerá (mas que permite inventar efeitos especiais na tela) é que, ao retornar aos caracteres de tamanho normal, tudo que estiver escrito na tela estará com um espaço em branco entre cada dois caracteres (pois cada caráter duplo ocupa o lugar de dois caracteres normais). Para verificar, rode

este exemplo e depois adapte esta dica às suas necessidades:

```
1 CLS
10 PRINT CHR$(23)
20 PRINT@80, "CARACTERES DUPLOS"
30 FOR I=1 TO 100 : NEXT
40 POKE 16912,40 : OUT 236,0 : REM - Mod III
50 FOR I=1 TO 100 : NEXT
60 GOTO 10
```

Roberto Quito de Sant'Anna-RJ

De azimute em rumo

Laci Mota Alves

O desenvolvimento de microcomputadores nacionais abre novas perspectivas de maior automatização da topografia no Brasil. Com relação aos micros da linha Sinclair, em decorrência de sua capacidade de memória de até 48 Kb de RAM e possibilidade de conexão a uma impressora, espera-se seja de grande valia na elaboração de planilhas topográficas. O que poderá causar incômodo em alguns momentos é a impressão dos somente oi-

tos dígitos significativos que, às vezes, são insuficientes para apresentar adequadamente certos resultados.

A calculadora programável HP-97, por sua vez, mostra-se eficiente na resolução de problemas de topografia. Tem a vantagem de imprimir até dez dígitos significativos, tem ótimos recursos de programação, mas apresenta só 26 registradores para o armazenamento de números e impressão em papel térmico. Entretanto, é bastante propícia à pre-

paração de dados a serem arquivados e processados pelo micro. Este, proporcionando o uso de caracteres alfanuméricos, imprime resultados mais bem elaborados, contribuindo para diminuir enganos e omissões a que estamos sujeitos.

Como exemplo, elaborou-se um programa de computação em BASIC em um TK82-C que transforma azimutes em rumos. Facilita sobremaneira a preparação de memoriais descritivos que constam de escrituras de compra e venda de propriedades rurais. Um memorial descritivo é feito de modo a fornecer a direção, o sentido e a distância de cada alinhamento que compõe a poligonal fechada que delimita a propriedade, assim como suas confrontações.

A direção e o sentido de cada alinhamento são dados, geralmente, pelo rumo magnético, que pode ser NORDESTE (NE), SUDESTE (SE), SUDOESTE (SW) ou NOROESTE (NW). Este programa poderá servir como sub-rotina para outro de natureza mais complexa e abrangente. Aceita azimutes no sistema sexagesimal e fornece resultados sob a forma G.MMSS Q, em que: G = grau, MM = minutos, SS = segundos, Q = quadrante (NE, SE, SW ou NW). Os azimutes maiores ou iguais a zero deverão ser digitados como G.MMSS, ex: 354.4532, que corresponde a 354º45'32".

Azimute

```
10 DIM R$(4,3)
11 DIM G$(1,7)
12 LET D=PI/PI+PI/PI
13 LET P=180
14 LET C=60
15 LET U=D/D
16 LET Z=10
20 LET R$(U)=" NE"
30 LET R$(D)=" SE"
40 LET R$(P/C)=" SW"
50 LET R$(D*D)=" NW"
51 LET DECIMAL=250
60 LET GRAU=310
70 PRINT "ENTRE COM AZIMUTE EM
G.MMSS"
71 PRINT "    RUMOS:";
80 INPUT AZ
90 LET G=AZ
120 GOSUB DECIMAL
140 IF G>P*D THEN LET G=G-INT
(G/P/D)*P*D
150 LET Q=INT (G*D/P)+U
160 IF Q>D+D THEN LET Q=U
170 LET R=((INT (Q/D)-G/P)/COS
(PI*Q))*P
180 LET G=R
210 GOSUB GRAU
211 LET G$(U)=" "
220 LET G$(U,D)=STR$ G
221 IF G>Z THEN LET G$(U,U TO
D)=STR$ G
222 LET G$(U,D+U)="."
223 LET G$(U,Z/D)=STR$ M
224 IF M>Z THEN LET G$(U,D*D T
O Z/D)=STR$ M
225 LET G$(U,Z/D+D)=STR$ S
226 IF S>Z THEN LET G$(U,C/Z T
O Z/D+D)=STR$ S
227 FOR I=D TO Z/D+D
228 IF G$(U,I)=" " THEN LET G$(
U,I)="0"
229 NEXT I
230 PRINT TAB 11;G$(U)+R$(Q)
231 GOTO 80
240 REM *****
250 REM *DECIMAL*
260 LET M=INT (G*Z*Z)/Z/Z
270 LET M=(M-INT M)/C*Z*Z
280 LET S=(G*Z*Z-INT (G*Z*Z))/P
/D*Z
290 LET G=INT G+M+S
300 RETURN
310 REM *GRAU*
320 LET MI=(G-INT G)*C
330 LET M=INT MI
340 LET S=INT ((MI-M)*C+U/D)
345 LET G=INT G
350 IF S=C THEN GOTO 370
351 IF M<C THEN GOTO P*D
352 LET M=U-U
353 LET G=G+U
360 RETURN
370 LET S=U-U
380 LET M=M+U
390 GOTO 351
```

Laci Mota Alves é engenheiro florestal formado em 1973 pela Universidade Federal do Viçosa - MG. É pós-graduado em Ciência Florestal e leciona atualmente Topografia e Fotogrametria na Escola Superior de Agronomia de Paraguaçu Paulista - ESAPP. Além disso, presta serviços autônomos de Topografia, Conservação do Solo e outros de Engenharia Florestal desde 1975.

Os melhores programas para você.



**Garantia
integral**



A Microsoft tem 120 programas em fitas e disquetes à sua disposição. São sistemas aplicativos para acompanhar e agilizar os negócios de sua empresa. E também jogos eletrônicos para você e sua família se divertirem muito. Todos especiais para TK-83, TK-85, TK-2000, Apple II e compatíveis. E todos com a mesma qualidade dos 100.000 programas já vendidos em todo o Brasil. Procure o revendedor Microsoft mais próximo (se não encontrar os programas Microsoft escreva para a Caixa Postal 54221 - CEP 01000-S. Paulo-SP). Você encontrará os melhores programas da sua vida.

MICROSOFT®
Sempre o melhor programa.

MICRO SISTEMAS, SEI — Sistemas Eletrônicos de Informações, divisão da Abril Cultural, e Telesp têm o prazer de apresentar...

TELESOFTWARE

o serviço Videotexto para usuários de microcomputadores.

Com o Telesoftware você tem à sua disposição, 24 horas por dia, em sua casa, uma ampla biblioteca de programas de uso pessoal, jogos e lazer, educativos, administrativos, financeiros, técnicos / científicos e utilitários publicados em MICRO SISTEMAS. São programas — a princípio compatíveis com a linha TRS-80 — selecionados e testados pela Revista, que você poderá carregar diretamente no seu micro, sem ter que digitá-los.

Para utilizar esse serviço, tudo o que você precisa é:

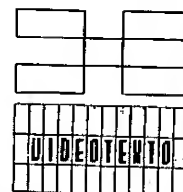
- ★ Ter um micro CP-500 ou CP-300
- ★ Ser assinante do Videotexto
- ★ Adquirir um kit composto de interface RS232-C, modem e software de comunicação

Se você tem um desses micros, não perca tempo. Entre em contato com a Telesp (Videotexto — Gerência de Usuários, tel.: 544-2535) e comece a utilizar esse novo e revolucionário meio de comunicação.

**Micro
Sistemas**



TELESP
TELECOMUNICAÇÕES
DE SÃO PAULO S.A.
EMPRESA DO SISTEMA TELEBRAS



PROGRAMAS INICIALMENTE DISPONÍVEIS

Programa	Autor	Publicado na Edição nº
Aprendendo Frações	Heber Jorge da Silva	30
Biorritmo	Francisco Luiz Farias Bezerra	26
Catálogo de Disquetes	Lawrence Falconer King	29
Ciclotron, o Jogo das Barreiras	Hendy Takeshi Yabiku	28
Compilador Forth	Antônio Costa	22
Copsys	Daniel Augusto Martins	29
Mini Editor de Textos	Ivan Camilo da Cruz	31
PIL — Controle da Fertilidade	Armando Oscar Cavanha Filho	31
Professor J. C.	Jôneson Carneiro de Azevedo	18
Registro Pessoal de Cheques	Marcelo Renato Rodrigues	25
Soletando	Heber Jorge da Silva	30
Tutor Matemático	Paulo Sergio Gonçalves	30

Todos os meses, este serviço será enriquecido com novos programas

**A I
CLUBE D**

Associe-se a
TIS, um exo
TOS de 10%
TAS, CURS
— Serviços
— Associaç
— Participa
Concurso
— Novos Lar
DOS (Bre
LHAS P/M
CLAIR -
nica")
ANUIDADE: O
samente ao re
um exemplar
Envie nome, e
seu anúncio
diária).

NO

Comunicaç
CAD
CEP 010

(*) Com
to-Soft

NO

TELE

O Lojão d
descomplic
Micros, pe
tos, Softwa
Revistas, C
nicos, Peça

PREQ
FINANCI

Compre pe
Reembolso

TELE

R. Marqu
Tel.: (0
(50000

(*) Sr. Indust
produto na
Contato en
220-7377 (i
disom)

**PRO
CO
ASSIN**

Escreva pa
mento de A
São Paulo e
a sua etique
sa.

Rio de Jane
Wilson, 165
RJ, CEP 2
262-5259 e
São Paulo
153, Jardim
01433 - tel
881-5668 e

M.S. Serviços

Alfa Bit CLUBE DE COMPUTAÇÃO

Associe-se ao ABCc e ganhe Anúncio GRATIS, um exemplar de ALFABIT e DESCONTOS de 10% na compra de LIVROS, REVISTAS, CURSOS e PROGRAMAS, além de:

- Serviços de "Reprinters" e Consultas
- Associação a Clubes Europeus
- Participação em Cursos, Congressos e Concursos
- Novos Lançamentos a PREÇOS REDUZIDOS (Breve: IMPRESSORA DE AGULHAS P/MINI-MICROS DE LÓGICA SINCLAIR - Lançamento "Digital Eletrônica")

ANUIDADE: Cr\$ 1.000 (hum mil cruzeiros) somente ao receber seu Cartão-Descontos e um exemplar de "Alfabit".

Envie nome, endereço, profissão e texto do seu anúncio (caso queira publicação imediata).

NOVIDÉIA(*)

Comunicação e Informática Ltda.
CAIXA POSTAL 9978
CEP 01051 - São Paulo, SP

(*) Comercializamos seu projeto-Soft ou Hard. Escreva-nos.

DATAMICRO

VENDA DE MICROCOMPUTADORES
TK 83, 85, & 2000 COLOR
CP 300, 500 & 600
COLOR 64 (EXT. BASIC)

SUPRIMENTOS
Disquete, fitas, form, contínuo

CONSULTORIA DE SISTEMAS
Diagnóstico e apoio à decisão

CURSOS E TREINAMENTO
Introdução aos microcomputadores

Linguagem Basic
Aplicação dos micros na Engenharia

Microcomputadores para crianças
INSCRIÇÕES ABERTAS

Livros e revistas especializadas

Visc. de Pirajá, 547 Sobreloja 211
Cep. 22.410 Ipanema Rio RJ
Tel.: (021) 274-1042
DESPACHAMOS PARA
TODO O BRASIL



MICROLÓGICA

Engenharia de Sistemas Ltda
Consultoria de Hardware
ASSISTÊNCIA TÉCNICA
A MICROCOMPUTADORES
Compatíveis com APPLE,
TRS80, IBM PC, ZX81, jogos
eletrônicos e outros.

Lançamento do contrato de
manutenção com custo
minimizado: 50% do valor
normal acrescido do custo
de peças (quando houver).

Temos programas comerciais,
utilitários, educacionais,
aplicativos e jogos novíssimos
para APPLE II em Diskettes
a partir de 15 mil
AV PRESIDENTE VARGAS, 542
/1912 — 263-9925
RIO DE JANEIRO (RJ)

PARA
PROBLEMAS
TÉCNICOS
USE
A CABEÇA



PARA PROBLEMAS COM MATERIAL DE

DESENHO - PINTURA - ENGENHARIA
PAPELARIA - ESCRITÓRIO - MÁQUINAS P/
ESCRITÓRIO E SUPRIMENTOS EM GERAL

O BEL-BAZAR
ELETRÔNICO

onde você AINDA encontra preço
e qualidade de ANTIGAMENTE!

AV. ALMIRANTE BARROSO, 61 - LJ "C"
TEL: 262-9229 - 262-9088 - 240-8410 - 221-8282
RIO DE JANEIRO - CASTELO

No Recife, visite

TELEVÍDEO (*)

O Lojão de Informática mais
descomplicado do país!
Micros, periféricos, suprimen-
tos, Software, Cursos, Livros e
Revistas, Componentes Eletro-
nicos, Peças e Som.

PREÇOS ESPECIAIS.
FINANCIAMENTO PRÓPRIO.

Compre pessoalmente ou pelo
Reembolso Postal:

TELEVÍDEO LTDA.

R. Marquês do Herval, 157
Tel.: (081) 224-8932,
(50000) Recife, PE

(*) Sr. Industrial: distribuímos seu
produto nas melhores condições.
Contato em São Paulo: Tel.: (011)
220-7377 (Sr. ANDERSON — Mun-
disom)

Sinclair Place

O lugar compatível
com você e seu
micro.

- Micros
- Acessórios
- Software
- Livros
- Revistas

Rua Dias da Cruz, 215
s/804 — Rio de Janeiro — RJ
Tel.: 594-2699

APPLE II

Transformação PAL-M
Assistência Técnica
Expansões

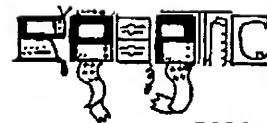
UNITRON

Assistência Técnica
Autorizada
Vendas • Leasing
Expansões

MICROEQUIPO

Manutenção
na sua empresa

Av. Marechal Câmara, 271/101
Rio de Janeiro Tel.: 262-3289



ROBOTIC

- MICROCOMPUTADORES
DE TODAS
AS MARCAS
- SUPRIMENTOS
- PEÇAS E PARTES PARA
MICROCOMPUTADORES
- JOGOS ELETRÔNICOS

RUA BARATA RIBEIRO, 370
— Loja 105 APART HOTEL —
COPACABANA — RIO — RJ
TEL: (021) 257-6396

Micro Sistemas

PROBLEMAS COM SUA ASSINATURA?

Escreva para o nosso Departa-
mento de Assinaturas do Rio ou
São Paulo e envie, para facilitar,
a sua etiqueta adesiva de remes-
sa.



Rio de Janeiro — Av. Presidente
Wilson, 165/grupo 1210, Centro,
RJ, CEP 20030 - tels.: (021)
262-5259 e 262-5208;
São Paulo — R. Oliveira Dias,
153, Jardim Paulista, SP, CEP
01433 - tels.: (011) 853-7758,
881-5668 e 853-3800.

LEIA

E

ASSINE

Micro
Sistemas

"MIKROS" AGORA NO LEBLON!

Av. Ataulfo de Paiva 566 - Loja 211
Rio de Janeiro — Tel.: 239-2798

APROVEITE OS PREÇOS
"INCRÍVEIS" DA "MIKROS"
DO LEBLON, APÓS SUA
ÉPOCA DE INAUGURAÇÃO.

MICROCOMPUTADORES

NAJA - JR-SYSDATA - UNITRON
COLOR 64 - APPLE-II PLUS
CP-200 - CP-300 - CP-500
TK-83 - TK-85 - RINGO

SISTEMAS

SOFTWARE (NAC. E IMPORT)
IMPRESSORAS E PERIFÉRICOS
CURSO DE BASIC

PROFISSIONAIS
ALTAMENTE ESPECIALIZADOS
PARA ATENDÊ-LO

BUTESHOP

INFORMATICA LTDA

PROGRAMAS PARA APPLE
E
COMPATÍVEIS

- Contabilidade Geral
- Editor de Texto
- Sistema de Arquivos
- Mala Direta
- Sistema Estatístico
- Controle Bancário
- Administração Imobiliária
- Sistema Bibliotecário
- Escritório de Advocacia
- Utilitários em Geral
- Jogos Diversos

DESENVOLVIMENTO DE SOFTWARES

- Atendimento pelo reembolso postal
- Vendas de computadores,
acessórios e suprimentos

Av. Franklin Roosevelt, 23/803 - Castelo
Tel.: (021) 220-6529 - 220-8327
C.E.P.: 20.021 Rio de Janeiro — RJ

MICRO BUG

O Comando M e as principais rotinas do SGM

Vamos iniciar este mês a construção do MICRO BUG, e nesse primeiro contato discutiremos algumas preocupações e cuidados a serem tomados. Em primeiro lugar, é preciso seguir à risca o texto e não *saltar* explicações ou procedimentos. É de vital importância que o leitor entenda o que está fazendo antes de se aventurar em modificações. Mesmo os leitores que já conhecem e dominam o Assembler devem ler com atenção cada tópico, pois tudo foi planejado para ter uma sequência tanto de funções quanto de explicações.

Outro ponto importante é quanto à gravação do MICRO BUG, mais precisamente quanto à fita cassete a ser empregada para tal. No tópico sobre os comandos de gravação iremos abordar esses problemas mais a fundo, porém no momento é suficiente o leitor separar uma fita exclusivamente para este trabalho. De preferência, uma fita C-46 de marca conhecida (*Scotch* ou *Basf*).

A REPRESENTAÇÃO DO ASSEMBLER Z80

O grande segredo para se dominar a linguagem de máquina está na capacidade de compreensão e interpretação dos mnemônicos da Z80. De fato, basta *entendemos* o que aqueles nomezinhos estranhos querem dizer para que todo o mistério desapareça.

Vejam um exemplo: o código 62 corresponde à instrução LD A, dd. Esta instrução serve para carregar o

LD BC,xxxx	====>	LET BC=xxxx
LD HL,zzzz	====>	LET HL=zzzz
LD (HL),dd	====>	POKE HL,dd
LD xxxx,(HL)	====>	POKE xxxx,L e POKE xxxx+1,H
LD A,(HL)	====>	LET A=PEEK HL
EX DE,HL	====>	LET HL=DE e LET DE=HL
JP xxxx	====>	GOTO xxxx (JP=Jump)
CALL zzzz	====>	GOSUB zzzz
LD D,dd	====>	LD D=dd
ADD A,C	====>	LET A=A+C
SBC HL,DE	====>	LET HL=HL-DE
ADD HL,BC	====>	LET HL=HL+BC
INC HL	====>	LET HL=HL+1
DEC DE	====>	LET DE=DE-1

Figura 1 - Algumas instruções Assembler possuem uma mecânica bastante semelhante a determinadas instruções do BASIC. Outras porém só são simuladas através de rotinas complexas.

acumulador com o valor genérico dd. Assim, poderíamos dizer que LD A,01 equivale a Load Acumulador com o valor 1. Em BASIC usaríamos o comando LET A = 1. Simples, não?

Como o acumulador, existem mais alguns registradores, porém só nos importarão os seguintes: H, L, B, C, D e E. Estes registradores são comumente agrupados na ordem HL, BC e DE, e desta forma qualquer um destes pares pode representar um valor entre 0 e 65.535 (2 bytes).

Na figura 1 temos algumas instruções do Assembler e os seus correspondentes aproximados em BASIC. A partir de agora usaremos as representações dd,xxxx,zzzz, etc. para indicar valores de 1 byte (dd,xx,yy,zz) e valores de 2 bytes (xxxx,dddd,yyyy,zzzz).

A FUNCIONALIDADE DO STACK DA Z80

Compreender o princípio de funcionamento do Stack da Z80 é, sem dúvida, o passo mais importante dentro da programação em linguagem de máquina. Acompanhe pela figura 2 a explicação a seguir: quando o sistema se organiza, ele determina um endereço para o par de registradores SP (*Stack Pointer*). Neste e nos endereços subsequentes, em ordem decrescente, a Z80 irá armazenar e recuperar temporariamente informações. Isso será feito pelas instruções PUSH e POP e cada informação armazenada ou recuperada deverá constar num par de registradores (HL, BC, DE ou AF). A posição do Stack é sempre atualizada em dois bytes e aponta para o primeiro

31997	xx	31997	xx	31997	xx
31998	xx	SP =>31998	zz	31998	zz
31999	xx	31999	dd	31999	dd
SP =>32000	xx	32000	xx	SP =>32000	xx
32001	xx	32001	xx	32001	xx

HL=ddzz ou	após PUSH HL	após POP DE
H=dd e L=zz		DE=ddzz

Figura 2 - O Stack (ou pilha da máquina, como é também conhecido) é a mais importante ferramenta da linguagem de máquina. Seu conhecimento e perfeito entendimento é de extrema importância para a criação de programas Assembler.

elemento da pilha e, assim, por exemplo, se o Stack estiver apontando para o endereço 32000, após um PUSH HL, o endereço 31999 conterá o valor do registrador H, o endereço 31998 conterá o valor de L e o Stack estará apontando para o endereço 31998.

Para recuperar estes valores basta fazer POP DE (por exemplo) que o registrador E assumirá o conteúdo do endereço 31998, o registrador D assumirá o conteúdo do endereço 31999 e o Stack estará novamente apontando para o endereço 32000. Note que POP não altera o conteúdo dos endereços, mas apenas modifica o valor do SP.

O Stack, no entanto, não é ocupado apenas dessa forma. Há uma outra forma de utilização que costuma causar uma infinidade de dores de cabeça aos programadores: através das sub-rotinas...

Imaginemos o nosso Stack no endereço 32000. Após uma instrução do tipo CALL xxxx, o Stack apontará para o endereço 31998 e os endereços 31999 e 31998 conterão o endereço de retorno da dita sub-rotina, ou seja, o endereço da primeira instrução após CALL xxxx. Quando houver uma instrução RET, o Stack apontará para o endereço 32000 e a execução do programa será desviada para o endereço armazenado em 31998 e 31999. Isso implica em que toda sub-rotina deve terminar por um RET, caso contrário o programa poderá ser desviado para um endereço desconhecido.

Obviamente, programar em linguagem de máquina não é só isso, e nesse momento seria prudente dar uma recapitulação no curso de Assembler da própria MICRO SISTEMAS.

OS PREPARATIVOS INICIAIS E A DIGITAÇÃO

A nossa primeira ação será reservar espaço no topo da RAM para que lá seja posicionado o SGM (Sistema Gerenciador de Módulos). Para o SGM iremos reservar 2 Kbytes, sendo que isso será feito pela alteração do valor da variável RAMTOP.

Faça POKE 16389, 120 e em seguida NEW. O comando NEW é fundamental, pois é ele quem organiza o Stack da Z80 (se for omitido, o Stack permanecerá ainda no topo da RAM). Quando, por um motivo ou outro, for dado um POKE dentro do Stack, o sistema perderá toda a referência de retorno e o micro muito provavelmente sairá do ar. Acompanhe pela figura 3 a mecânica da reserva de espaço (lembre-se sempre que toda vez em que houver necessidade de reservar espaço no topo da RAM o procedimento deverá ser o que acabamos de ilustrar, ou seja, POKE e NEW).

Passemos então à listagem 1. Ela servirá para a introdução da primeira parte do SGM, com a impressão dos endereços em hexadecimal, pois este será o padrão adotado daqui para frente. Após a digitação da listagem 1, digite RUN e responda à pergunta ENDEREÇO? com o primeiro endereço da listagem ou bloco que for digitar. Os endereços subsequentes não precisam ser digitados.

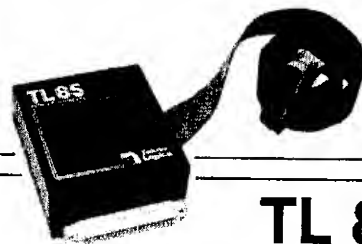
Digitando M, o sistema retorna à pergunta ENDEREÇO?, e digitando Z o

organização inicial	SP	RAMTOP=32768
memória RAM		
após POKE 16389,120	RAMTOP=30720	SP
memória RAM	área a ser reservada	
após NEW	SP	RAMTOP=30720
memória RAM	área reservada	

Figura 3 - Todo micro (sistema operacional, BASIC, rotinas em Assembler etc) ocupa o Stack o tempo todo. A sua manipulação deve ser cercada de cuidados, pois qualquer deslize ocasiona a perda total de controle sobre o sistema. Este, aliás, foi o grande problema encontrado pelos leitores de MICRO SISTEMAS quando tentaram digitar o Monitor BASIC (MS número 25) sem antes alterar o valor do Stack Pointer (SP) através do comando NEW.

Era só o que faltava...

Agora, com as interfaces TL 85 e TL 300, você já pode ligar o seu micro TK 85, TK 83 e CP 300 em uma impressora.



TL 85

Interface que complementa seu TK 85 ou TK 83, ampliando largamente a aplicação do mesmo.

- aciona qualquer impressora ou máquina de escrever eletrônica, com comunicação paralela Centronics;
- aceita todos os comandos relacionados com impressora (Copy, Llist, Lprint);
- gera maiúsculas, minúsculas, acentos e controles do ASCII, diretamente do teclado;
- permite a conexão de outras expansões.

Versão com Editor: Facilímo de usar; orientado para língua portuguesa; gravado em EPROM; marginação automática; separação silábica e outros recursos.



TL 300

Acionada pelos comandos de impressão, a interface TL 300 possibilita a ligação do CP 300 a qualquer impressora ou máquina de escrever eletrônica, com comunicação paralela Centronics. Permite, ainda, a expansão do sistema, oferecendo condições para que o micro aceite outros periféricos.



Produtos e Serviços para a Informática Ltda.

Av. São Pedro, 1062 - Fone (0512) 42-8549
90000 Porto Alegre - RS

Distribuidores

São Paulo: Pró-Controle Com. e Controle Ltda. - Fone (0192) 32-7364 - Campinas
Goiás e Distrito Federal:
New Computadores Comércio e Representações Ltda.
Fone (061) 274-5060 - Brasília

Listagem 1

```

5 FAST
10 SCROLL
15 SCROLL
20 PRINT "ENDEREÇO?"
25 INPUT E$
30 LET E=CODE E$*4096+CODE E$(
2)*256+CODE E$(3)*16+CODE E$(4)-
122332
35 LET US=""
40 SCROLL
45 PRINT E$;" ";CHR$(INT (P
EEK E/16)+28)+CHR$(PEEK E-INT (
PEEK E/16)*16+28)+""
50 IF US="" THEN INPUT US
55 IF US="M" THEN GOTO 10
56 IF US="Z" THEN GOTO 100
57 IF US="" THEN GOTO 75
60 PRINT US( TO 2)
65 POKE E, CODE US*16+CODE US(2
)-476
70 LET US=US(3 TO )
75 LET E=E+1
80 LET A=INT (E/256)
85 LET B=E-A*256
90 LET E$=CHR$(INT (A/16)+28)
+CHR$(A-INT (A/16)*16+28)+CHR$(
INT (B/16)+28)+CHR$(B-INT (B/1
6)*16+28)
95 GOTO 40
100 CLEAR
105 DIM A$(2048)
110 FOR A=1 TO 2048
115 LET A$(A)=CHR$(PEEK (A+3071
9))
120 NEXT A
125 PRINT AT 21,0;"PRONTO PARA
GRAVAR"
130 PAUSE 4E4
135 SAVE "MICRO BUG"
140 FOR A=1 TO 2048
145 POKE A+30719, CODE A$(A)
150 NEXT A

```

sistema grava na fita cassete uma cópia do SGM, ou de tudo que já houver sido digitado.

Os blocos Assembler são de digitação direta, porém as listagens em mnemônicos requerem um cuidado maior (veja a listagem 2). Elas são constituídas, na maioria das vezes, por quatro colunas. A primeira (à esquerda) corresponde aos endereços em hexadecimal onde estão (ou estarão) os códigos de máquina. Na segunda coluna estão os códigos propriamente ditos e na terceira coluna estão os mnemônicos correspondentes a estes códigos. A quarta coluna constitui uma opção para comentários e estes serão sempre referenciados por um ponto e vírgula.

Para digitar a listagem em mnemônicos, responda à pergunta **ENDEREÇO?** com o primeiro endereço da coluna 1 e a seguir introduza os códigos da coluna 2.

Toda vez que for digitar códigos de máquina, em blocos ou em listagens, não esqueça de fazê-lo com a maior atenção possível, pois qualquer erro será invariavelmente fatal. É preferível perder-se um pouco mais de tempo na digitação do que procurar um código errado.

Ainda com relação à listagem 2, note que em ambas as rotinas o valor original do par HL é preservado. O **SCROLL** para cima ajusta a próxima impressão em AT 21,0 e o **SCROLL** para baixo em AT 0,0. Observação: nos comentários, algumas instruções **LET** e **THEN** são omitidas para simplificar a explicação; todos os valores, salvo os endereços do próprio SGM, são apresentados em decimal.

A APRESENTAÇÃO E A IMPRESSÃO DE STRINGS

Todo programa que pretenda um mínimo de clareza deve conter alguns elementos que o identifiquem e criem uma distinção em relação a outros programas. Além disso, é importante manter a fonte, pois pode-se necessitar, no futuro, de informações acerca do sistema. Se a origem do programa for omitida, poderão ocorrer problemas de informação difíceis de serem solucionados. O crédito do autor também não deve ser substituído pelo nome do usuário, pois tal atitude não resulta em nenhum ganho para o seu praticante, a não ser tal-

vez a satisfação de ver seu nome na tela do vídeo.

O SGM possui uma apresentação bastante sofisticada, porém sua mecânica é simples. O sistema todo funciona através de **SCROLL**, tanto para baixo como para cima (veja listagem 2), porém a principal linha é sempre a da posição 21. Quando se entra no sistema, ou quando se produz um **RESET**, é apresentado o nome **MICRO BUG**, seguido da fonte **MICRO SISTEMAS** e logo abaixo o nome do módulo, se houver algum implementado. A posição AT 21,0 contém o **prompt** ">" e o cursor (**GRAPHICS** 4).

Os comandos funcionam diretamente, sem *label*, e cada tecla corresponde a um comando. As teclas de S a Z são reservadas para o módulo que estiver implementado.

A impressão do nome **MICRO BUG**, em caracteres gráficos, será detalhadamente discutida quando virmos a impressão de valores numéricos, pois ela utiliza, através de um macete, a rotina de impressão de dois dígitos hexadecimais. O princípio de impressão das

Listagem 2

rotina \$SCRUP (scroll para cima)

7860 E5	PUSH HL	;arquivo HL
7861 2A 0C 40	LD HL,(400C)	;HL=PEEK 16396+256*PEEK 16397
7864 E5	PUSH HL	
7865 11 21 00	LD DE,0021	;DE=33
7868 19	ADD HL,DE	;HL=HL+DE
7869 D1	POP DE	
786A 01 D6 02	LD BC,02D6	;BC=726
786D E0 80	LDIR	;produz o scroll
786F 0E 41	LD C,41	;C=65
7871 A7	AND A	
7872 ED 42	SBC HL,BC	;HL=HL-BC
7874 22 0E 40	LD (400E),HL	;POKE 16398,L e POKE 16399,H
7877 E1	POP HL	;recupera HL
7878 C9	RET	;RETURN

rotina \$SCRDN (scroll para baixo)

7879 E5	PUSH HL	;arquivo HL
787A 2A 0C 40	LD HL,(400C)	;HL=PEEK 16396+256*PEEK 16397
787D E5	PUSH HL	
787E 01 B4 02	LD BC,02B4	;BC=692
7881 09	ADD HL,BC	;HL=HL+BC
7882 11 21 00	LD DE=0021	;DE=33
7885 E5	PUSH HL	
7886 19	ADD HL,DE	;HL=HL+DE
7887 D1	POP DE	
7888 E8	EX DE,HL	;inverte valores DE e HL
7889 ED B8	LDDR	;produz scroll
788B AF	XOR A	;A=0
788C 06 20	LD B,20	;B=32
788E 23	INC HL	;HL=HL+1
788F 77	LD (HL),A	;POKE HL,A
7890 10 FC	DJNZ 788E	;B=B-1 : IF B<>0 GOTO 788E
7892 E1	POP HL	
7893 23	INC HL	;HL=HL+1
7894 18 DE	JR 7874	;GOTO 7874

Listagem 3

rotina \$PRINT (impressão de strings)

```

78C1 CD 60 78      CALL 7860      ;GOSUB $SCRUP
78C4 CD 60 78      CALL 7860      ;GOSUB $SCRUP
78C7 E1            POP HL         ;HL=ender do carac
78C8 7E            LD A,(HL)      ;A=PEEK HL
78C9 E6 7F         AND 7F         ;IF A>128 THEN LET A=A-128.
78CC D7            RST 10         ;PRINT CHR$ A;
78CC CB 7E         BIT 7,(HL)     ;IF PEEK HL>128 THEN LET Z=0
78CE 23            INC HL         ;HL=HL+1
78CF E5            PUSH HL        ;repop STACK
78D0 C0            RET NZ         ;IF Z=0 THEN RETURN
78D1 18 F4         JR 78C7        ;GOTO 78C7

```

para imprimir uma string (por exemplo PRINT "MICRO SISTEMAS") o procedimento deve ser o seguinte:

```

xxxx CD C1 78 ..   CALL 78C1
xxxx+3 32 2E 28 37 define MICRO SISTEMAS
      34 00 38 2E
      38 39 2A 32
      26 B8
xxxx+17 11 ...     LD ...

```

strings, por outro lado, baseia-se na sua definição logo após uma chamada CALL, o que em BASIC equivaleria a PRINT "texto".

Isso é possível pela manipulação do Stack da Z80 (listagem 3). Logo após a chamada, o sistema obtém o endereço de retorno, ou seja, o endereço após CALL, e passa a imprimir como caracteres os bytes subsequentes. A cada byte impresso, o Stack é atualizado para que, no final, o sistema saiba para onde voltar. A sinalização de final de string é feita pelo bit 7 do último caráter. Se ele estiver setado (= 1), então o sistema retorna para a instrução seguinte à string.

Este artifício é usado pelo SGM para imprimir todas as mensagens do sistema e pode ser facilmente alterado para outros usos. A maior característica dele é que, na string definida, o último caráter deve ser sempre em vídeo inverso (repare, também na listagem 3, que o último S de SISTEMAS deve ser em caráter inverso, e que após a impressão da string o programa retorna para executar a instrução em xxxx+17).

TABELA DE COMANDOS, RESET E A ENTRADA NO MICRO BUG

Para que o sistema responda às teclas pressionadas com a execução de um comando é preciso que haja algum tipo de processo do gênero: IF INKEY\$ = "B" THEN GOTO...

Na realidade, o processo utilizado no SGM é bastante diferente deste, ou seja,

definiremos uma tabela de endereços que corresponda a cada uma das teclas. Nosso intervalo válido será de A a Z, sendo que qualquer outra tecla deverá ser rejeitada pelo sistema.

A tabela propriamente dita inicia no endereço 7940 e vai até o endereço 7973. Cada dois bytes possuem o endereço de execução de cada comando (veja listagem 4).

Quando se entra no MICRO BUG, ou após um RESET, ou quando termina a execução de um comando, o sistema fica aguardando que uma tecla/comando seja pressionada. Se para aquela tecla não houver comando, o sistema informa com a mensagem COMANDO INEXISTENTE. A este estado daremos o nome de Loop de Comando. (Na listagem 4, observe que os comandos de S a Z são implementados automaticamente pelo módulo específico. O endereço 7A34 equivale a COMANDO INEXISTENTE e deve constar de todas as letras cujos comandos ainda não estejam no SGM).

O Loop de Comando (listagem 5) é constituído por uma rotina bastante simples e que funciona da seguinte for-

Listagem 4 — Tabela de Definição dos Comandos

7940	34 7A	def 7A34	;comando A
7942	34 7A	def 7A34	;comando B
7944	34 7A	def 7A34	;comando C
7946	34 7A	def 7A34	;comando D
7948	34 7A	def 7A34	;comando E
794A	34 7A	def 7A34	;comando F
794C	34 7A	def 7A34	;comando G
794E	34 7A	def 7A34	;comando H
7950	34 7A	def 7A34	;comando I
7952	34 7A	def 7A34	;comando J
7954	34 7A	def 7A34	;comando K
7956	34 7A	def 7A34	;comando L
7958	34 7A	def 7A34	;comando M
795A	34 7A	def 7A34	;comando N
795C	34 7A	def 7A34	;comando O
795E	34 7A	def 7A34	;comando P
7960	34 7A	def 7A34	;comando Q
7962	34 7A	def 7A34	;comando R
7964	34 7A	def 7A34	;comando S
7966	34 7A	def 7A34	;comando T
7968	34 7A	def 7A34	;comando U
796A	34 7A	def 7A34	;comando V
796C	34 7A	def 7A34	;comando W
796E	34 7A	def 7A34	;comando X
7970	34 7A	def 7A34	;comando Y
7972	34 7A	def 7A34	;comando Z

Listagem 5 — Loop de Comando

79EA	ED 73 3E 79	LD (793E), SP	;arquivo SP para reset
79EE	CD 84 79	CALL 7984	;imprime título e fonte
79F1	CD 2B 0F	CALL 0F2B	;implementa SLOW
79F4	21 27 79	LD HL, 7927	;verifica se ha' modulo
79F7	CB 56	BIT 2, (HL)	implementado
79F9	C4 10 78	CALL NZ, 7810	
79FC	ED 78 3E 79	LD SP, (793E)	;recupera SP
7A00	3E 12	LD A, 12	;prompt ")"
7A02	D7	RST 10	
7A03	2A 0E 40	LD HL, (400E)	
7A06	36 04	LD (HL), 04	;cursor
7A08	CD AA 78	CALL 78AA	;identifica tecla pressionada
7A0B	CD 03 78	CALL 78D3	se houver
7A0E	FE 77	CP 77	;ignore se for DELETE
7A10	28 F6	JR Z, 7A08	
7A12	FE 76	CP 76	;ignore se for ENTER
7A14	28 F2	JR Z, 7A08	
7A16	D7	RST 10	
7A17	CD AA 78	CALL 78AA	;imprime tecla do comando
7A1A	D6 26	SUB 26	;espera tecla ser liberada
7A1C	38 16	JR C, 7A34	;calcula valor da tecla
7A1E	FE 1A	CP 1A	;tecla fora da faixa
7A20	3F	CCF	imprime COMANDO INEXISTENTE
7A21	38 11	JR C, 7A34	
7A23	17	RLA	
7A24	06 00	LD B, 00	;multiplica tecla por 2
7A26	4F	LD C, A	;BC= valor modulo para a
7A27	21 40 79	LD HL, 7940	TABELA DOS COMANDOS
7A2A	09	ADD HL, BC	;base da tabela
7A2B	5E	LD E, (HL)	
7A2C	23	INC HL	;DE= ender de execucao do
7A2D	56	LD D, (HL)	comando
7A2E	21 4A 7A	LD HL, 7A4A	
7A31	E5	PUSH HL	;prepara retorno e execucao
7A32	05	PUSH DE	do comando
7A33	C9	RET	
7A34	CD C1 78	CALL 78C1	;executa comando
7A37	28 34 32 26		;imprime mensagem
	33 29 34 00		COMANDO INEXISTENTE
	2E 33 2A 3D		
	2E 38 39 2A		
	33 39 AA		
7A4A	CD 60 78	CALL 7860	;scroll para cima
7A4D	CD 60 78	CALL 7860	;scroll para cima
7A50	18 AA	JR 79FC	;aguarda novo comando

ma: os comandos são chamados como se fossem sub-rotinas. Para o retorno ao *Loop de Comando*, basta haver uma instrução RET, mas em comandos mais complexos pode-se lançar mão de um RESET que, dependendo do endereço de JUMP, vai produzir um resultado diferente.

No endereço 79EE o RESET é total, havendo a impressão do título, fonte e módulo implementado. No endereço 79FC o RESET não produz alterações na tela de vídeo, o que pode causar alguma confusão com o cursor. No endereço 7A4A o RESET produz dois SCROLL para cima, repondo o cursor na posição original 21,0. Observação: na listagem 5, note a forma como foi encontrado o endereço de execução do comando e o modo como ele foi chamado (o RET do endereço 7A33 equivale a um GOTO, pois usa a manipulação do Stack como forma de desvio controlado).

AS ROTINAS OPERACIONAIS DO SGM

Existem ainda outras rotinas operacionais no SGM, mas não as veremos detalhadamente agora uma vez que algumas são bastante complexas e necessitam de um certo conhecimento de linguagem de máquina. Mais na frente iremos retornar a elas, mas para os que já conhecem o Assembler a figura 4 contém uma lista com os respectivos nomes e endereços. As rotinas estão na listagem 6 e devem estar no SGM antes de fazermos a nossa primeira entrada nele.

O COMANDO B

Quando fizermos a nossa primeira entrada no MICRO BUG, devemos ter o cuidado de ter implementado o comando B, pois será ele quem produzirá o retorno ao BASIC. A utilização deste

Listagem 6 — Rotinas do SGM

783C	00 C9 F5 1F 1F 1F 1F CD
7844	47 78 F1 F5 E6 0F E5 21
784C	27 79 C8 46 E1 20 09 C8
7854	5F 28 02 EE 88 D7 F1 C9
785C	C6 1C 18 F9

7896	D6 DE 47 0E 01 28 04 CB
789E	11 10 FC 3A 26 79 A9 32
78A6	26 79 3E FF F5 CD BB 02
78AE	7C FE FE 38 F8 06 03 18
78B6	03 F5 06 18 AF 0B 88 20
78BE	FC F1 C9

78D3	CD BB 02 7C FE FE 30 F8
78D8	44 4D CD 8D 07 7E FE 76
78E3	C8 FE 77 C8 FE CD 28 1D
78EB	FE 74 28 24 FE DE 04 96
78F3	78 FE 75 CA EE 79 00 00
78FB	00 00 00 00 00 00 00 00
7903	CB 77 C8 18 CB CD 2A 0A
790B	01 01 15 CD F5 08 18 CD
7913	3A 27 79 EE 10 32 27 79
7918	18 B6

7926	00 01 00 00 00 00 00 00
792E	00 00 00 00 00 00 00 00
7936	00 00 00 00 00 00 00 00
793E	00 00

7974	80 F4 8B 80 8B 4F 8B 40
797C	8B 40 FB 84 F4 08 0B 84
7984	9A 84 08 0D 50 3F 50 8D
798C	50 80 F5 08 D5 08 D5 03
7994	82 D5 08 0D 50 00 78 1D
799C	50 80 D7 3A D5 08 D5 FB
79A4	80 D5 88 82 AB 7D 52 A2
79AC	AB 7D 0A 87 2A 87 2A 87
79B4	CD 2A 0A 21 27 79 C8 86
79BC	2E 74 06 40 7E CD 3E 78
79C4	23 10 F9 2E 27 C8 C6 CD
79CC	C7 78 32 2E 28 37 34 00
79D4	38 2E 38 39 2A 32 26 38
79DC	00 17 1D 25 24 20 97 01
79E4	00 15 CD F5 08 C9

7A52	2A 0E 40 36 04 CD AA 78
7A5A	CD D3 78 2A 0E 40 36 00
7A62	28 FE 76 20 02 7E C9 FE
7A6A	77 20 1A 7E FE 76 22 0E
7A72	40 20 24 CD C1 78 29 2A
7A7A	31 2A 39 2A 00 2E 31 2A
7A82	2C 26 B1 18 C3 E5 21 27
7A8A	79 C8 66 E1 28 02 C8 FF
7A92	07 23 23 7E FE 76 C8 CD
7A9A	AA 78 18 B4 01 03 15 CD
7AA2	F5 08 2A 0E 40 11 00 00
7AAA	7E FE 0D 28 20 00 7E B7
7AB2	C8 FE 1A C8 D6 1C 38 10
7ABA	FE 10 30 0C E8 06 04 29
7AC2	10 FD 4F 09 E8 23 18 E6
7ACA	CD C1 78 29 2E 2C 2E 39
7AD2	34 00 2E 31 2A 2C 26 B1
7ADA	18 A9 23 7E B7 C8 FE 1A
7AE2	C8 D6 1C 38 E3 FE 0A 30
7AEA	0F E5 F5 21 00 00 3E 0A
7AF2	06 08 29 17 30 01 19 10
7AFA	F9 F1 4F D9 E8 E1 18 DA
7B02	3A 26 79 C8 5F CC 87 78
7B0A	C8 57 20 0D 7C CD 3E 78
7B12	7D CD 3E 78 AF D7 D7 D7
7B1A	C9 E5 1E FF 01 F0 08 CD
7B22	E1 07 01 18 FC CD E1 07
7B2A	01 9C FF CD E1 07 0E F6
7B32	CD E1 07 7D CD E8 07 2A
7B3A	0E 40 06 09 7E 05 28 FE
7B42	76 20 F9 AF D7 10 FD E1
7B4A	C9

comando é direta, ou seja, quando o sistema estiver no *Loop de Comando* bastará pressionar a tecla B. Para a implementação, faça as modificações da listagem 7.

O COMANDO M

O comando M será detalhado nos próximos tópicos pois requer uma série de cuidados para a sua compreensão — no momento estamos muito mais interessados em ver a coisa funcionar do que na teoria propriamente dita, certo? A listagem 8 contém todo o comando que deve ser implementado da mesma forma que o comando B (não esquecer de alterar a *Tabela de Comandos*).

O funcionamento do comando M possui, de certa forma, toda a mecânica do sistema MICRO BUG. Ele é o principal recurso do SGM, pois permite que toda a memória do micro seja vasculhada e que, na RAM, possa haver modificações. Lembre-se, no entanto, que não é o comando quem altera ou modifica os dados e sim o operador; portanto, antes de fazer qualquer coisa convém saber o que se está fazendo.

No *Loop de Comando*, ao ser pressionada a tecla M, o sistema imprime M, pula um espaço e passa a esperar um endereço. Se a tecla ENTER for pressionada, o sistema utilizará o último endereço utilizado pelo comando M anterior. Caso contrário, o que for digitado será interpretado como um endereço em hexadecimal. Após a digitação de um endereço, o sistema o imprimirá na última linha, seguido de seu conteúdo em hexadecimal.

Nesse estágio, muitas opções podem ser utilizadas. São elas: tecla K (+) (avança um endereço); tecla J (-) (retrocede um endereço); tecla BREAK (interrompe o comando e retorna ao *Loop de Comando*); e teclas 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F (são interpretadas como um dado em hexa-

```
$BYTE(783E)  rotina de impressao de um valor em hexadecimal
$CONTR(7894)  rotina de manipulacao dos bits de MFLAG
$CONT(78AA)   rotina de espera ate que a tecla pressionada seja liberada
$TEMPO(78B7)  retarda a execucao do sistema por algum tempo
$KEY(78D3)    espera por uma tecla pressionada
$CLS(798)     produz um CLS no video
$GRAPH(7910)  modo GRAPHICS de $CURSOR
$CURSOR(7A52) rotina do cursor
$VALDE(7A9E)  obtem um valor numerico digitado pelo usuario
$ENDER(7B02)  imprime um endereco
```

Figura 4

decimal para modificar o conteúdo do endereço na posição AT 21,0).

ENTRANDO NO MICRO BUG

Se você leu o texto com atenção, seguiu à risca as instruções e já digitou tudo, tanto os blocos Assembler quanto as listagens em mnemônicos, então está na hora de entrarmos no SGM e verificarmos se, na prática, a teoria funciona.

Em primeiro lugar, faça uma cópia do SGM, pois ainda não sabemos se tudo funciona direitinho. Em segundo lugar, dê um NEW, pois a partir de agora, para continuarmos a digitar, iremos utilizar o próprio SGM. Dê então um RAND USR 31210 e lá vamos nós...

Se correu tudo bem, no alto da tela irá aparecer MICRO BUG em letras garrafais, a fonte MICRO SISTEMAS e o cursor no canto esquerdo do vídeo. Não vacile agora e teste o comando B como

primeira providência, pois não queremos que o micro *perca o rumo de casa*.

Retorne ao MICRO BUG e vamos então ao comando M. Digite M e, como o sistema deve estar apontando para a ROM, vamos digitar um endereço qualquer (na falta de um melhor serve 4082). O resultado deve ser instantâneo, caso contrário deve existir erro em algum lugar.

Teste a tecla K(+). Pressione-a por algum tempo e espere a tela ficar cheia. Agora pressione J(-) e veja o resultado. Vamos fazer outro teste: digite SHIFT S e retorne ao teste das teclas K e J. Veja a diferença, pois SHIFT S chaveia a velocidade de impressão (se pressionada novamente, retorna à velocidade lenta). Digite agora SHIFT 3 e o conteúdo dos bytes será apresentado como um valor decimal (SHIFT 3 retorna à condição inicial). SHIFT E faz a mesma coisa, porém com o endereço.

Listagem 7 — Comando B

```
7942 1D 79          def 791D          ;comando B

791D CD C7 78       CALL 78C7         ;imprime sem scroll
7920 26 38 2E       ;ASIC
      AB
7924 E1             POP HL            ;limpa STACK
7925 C9             RET              ;retorna ao Basic
```

OS ANJOS DA GUARDIAN

Proteção integral para o seu Micro

Estabilizador Eletrônico

mini REG

Proteção ultra-rápida contra variações da tensão da rede em até ± 22% estabilizando-a em ± 1%. Capacidade de 0.25, 0.4, 0.6, 0.8 e 1 KVA.



Gerador Eletrônico

GERATRON

Quando a rede elétrica faltar, GERATRON continuará alimentando o seu micro como se nada houvesse acontecido. Capacidade de 200 VA continuamente e 500 VA de pico. Bateria interna com autonomia de 90 minutos a plena carga.



NO-BREAK

Linha Especial para Micros

Proteção completa para o seu micro, mantendo a alimentação altamente estável e sem interrupção. Forma de onda senoidal. Capacidades de 0.25, 0.4, 0.6, 1, 1.5, 2.5, 3.5 e 5 KVA. Opera com quatro baterias comuns de 12 volts.

GUARDIAN
EQUIP. ELETRON. LTDA.



R. Dr. Garnier, 579 - CEP 20971 - Rocha - Telex: PABX (021) 261-6458 - Direto 201-0195 - Telex nº (021) 34016 - Rio de Janeiro - RJ
Representante São Paulo - Tel.: (011) 270-3175 - Representantes em todas as capitais.

Como já vimos, os conteúdos podem ser modificados, bastando que o usuário digite um dígito hexadecimal. Essa condição pode ser alterada por **SHIFT D**, o que faz o sistema aceitar um valor decimal. A entrada em decimal, por outro lado, requer a tecla **ENTER** para validar as modificações.

Na realidade, o **MICRO BUG** funciona com números tanto em decimal quanto em hexadecimal. A sua condição normal é, de preferência, em hexadecimal, por serem números de fácil manipulação. No entanto, se o usuário desejar introduzir um endereço em decimal, basta anteceder o valor pelo especificador **\$** (desse modo, **4082** em hexadecimal equivale a **\$16514** em decimal). Essa condição também poderá ser alterada, mas isso será visto mais adiante. Vale lembrar que a partir de agora usaremos o próprio **SGM** para as novas implementações.

Este projeto vem sendo desenvolvido pela equipe do CPD de MICRO SISTEMAS, sob a coordenação de Renato Degiovani.

Listagem 8 — Comando M

7958	7B 7B	def 7B7B	; comando M
7B4B	2A 2D 79 23 22 2D 79 CD		
7B53	60 78 CD 02 7B 7E 21 26		
7B5B	79 CB 46 2D 03 C3 3E 7B		
7B63	4F AF 47 C3 98 0A 2A 2D		
7B6B	79 2B 22 2D 79 01 15 00		
7B73	A7 ED 42 CD 79 78 18 DA		
7B7B	AF D7 CD 52 7A B7 28 0B		
7B83	CD 9E 7A EB 2B 22 2D 79		
7B8B	CD 60 78 CD 4B 7B CD D3		
7B93	7B B7 C8 FE 30 2B F4 FE		
7B9B	1C 3B F3 FE 2C 3B 07 FE		
7BA3	2F CC 69 7B 18 EB F5 01		
7BAB	0B 15 CD F5 0B 3A 26 79		
7BB3	CB 77 2D 29 F1 D7 CD AA		
7BBB	7B D6 1C A7 17 17 17 17		
7BC3	32 7B 40 CD D3 7B FE 1C		
7BCB	3B F9 FE 2C 30 F5 D7 D6		
7BD3	1C FD 86 7B 2A 2D 79 77		
7BDB	CD AA 7B 1B AE F1 D7 CD		
7BE3	52 7A 01 0B 15 CD F5 0B		
7BEB	2A DE 4D 11 00 00 CD DD		
7BF3	7A 7B 1B E0		

ATENÇÃO LOJAS E REPRESENTANTES

Novas lojas surgem em todas as partes do país, enquanto as já existentes ampliam seus negócios e diversificam suas linhas de produtos e serviços. Elas são o termômetro deste incrível mercado.

Para acompanhar esse acelerado crescimento a nível nacional, **MICRO SISTEMAS** está precisando do auxílio destas empresas, mediante o fornecimento de informações básicas sobre suas atividades.

Portanto, se você é Diretor ou Gerente de uma Loja, ou Revendedor de sistemas ou suprimentos de microcomputadores, mantenha o nome de sua empresa presente e atualizado em nosso cadastro de informações, mandando-nos os

seguintes dados:

- Razão Social
- CGC
- Endereço e Telefone
- Produtos que comercializa (hardware, software, publicações, móveis etc)
- Nome dos Diretores/Gerentes
- Um breve histórico da firma

E quaisquer informações que julgar importantes.

Estes dados vão nos ajudar a fazer uma cobertura mais completa do mercado brasileiro de microinformática, aumentando sua divulgação e, eventualmente, transformando em notícia o nome de sua empresa.

Redação de MICRO SISTEMAS

Av. Presidente Wilson, 165/grupo 1210, Centro, Rio de Janeiro, RJ, CEP 20030

Tels.: (021) 262-5259



SOLUÇÃO NÃO É PROBLEMA



não importa o tamanho de seu problema,
nós temos a solução na medida exata!

CP-200 COM
SPEED



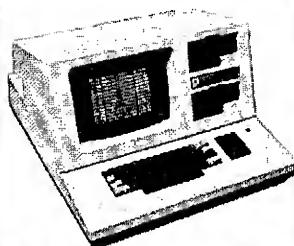
- LINGUAGEM BASIC
- 16 K DE MEMÓRIA
- VELOCIDADE DE TRANSFERÊNCIA 14 VEZES MAIS RÁPIDA

CP-300



- MODULAR
- LINGUAGEM BASIC
- 48 K DE MEMÓRIA
- COMPATÍVEL COM SOFTWARE DO CP-500

CP-500



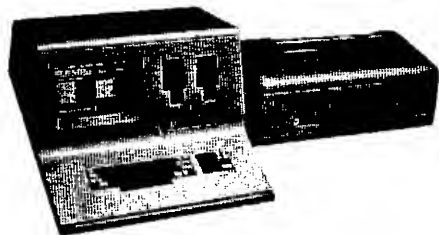
- LINGUAGEM BASIC
- 48 K DE MEMÓRIA
- ATÉ 4 DRIVES
- SAÍDA PARALELA SERIAL

P-500



- VELOCIDADE 100 CPS
- MATRIZ 9 x 7
- INTERFACE: PARALELA SERIAL

S-600



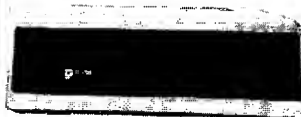
MICRO:

- LINGUAGENS COBOL, BASIC E FORTRAN
- 64 K DE MEMÓRIA
- DUAS UNIDADES DE DISCO

IMPRESSORA:

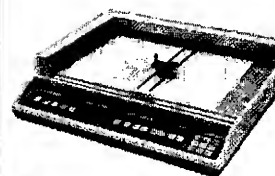
- VELOCIDADE 130 CPS
- MATRIZ 7 x 9
- 132 COLUNAS
- ORIGINAL + 5 CÓPIAS

P-720



- VELOCIDADE 200 CPS
- MATRIZ 7 x 9
- INTERFACE: PARALELA, SERIAL

TRAÇADOR
GRÁFICO



- 8 PENAS
- ÁREA DE TRAÇADO 10 x 15 POL.
- INTERFACE RS-232

ACESSÓRIOS

SOFTWARE • MESAS • DISQUETES • ARQUIVOS • FORMULÁRIOS CONTÍNUOS • ESTABILIZADORES DE TENSÃO • UNIDADES DE DISCO FLEXÍVEL • ETC.

APROVEITE!

PROMOÇÕES ESPECIAIS • FINANCIAMENTO • LEASING • CONSÓRCIO • CARTÕES DE CRÉDITO: CREDITCARD, NACIONAL, ELLO.

filcrest

Filcrest Importação e Representações Ltda.

Rua Aurora, 165 - CEP 01209 - São Paulo - SP

Telex 1131298 FILG BR - PBX 223-7388 - Ramais 2, 4, 12, 18, 19 - Diretos: 223-1446, 222-3458, 220-5794 e 220-9113 - Reembolso - Ramal 17 Direto: 222-0016 - 220-7718

Outras palavras em FORTH

Ivan Camilo da Cruz

Em MICRO SISTEMAS número 22, foi apresentado um compilador para a linguagem FORTH de autoria de Antônio Costa. Este artigo introduziu aos leitores desta revista esta fantástica linguagem, que consegue ser ao mesmo tempo simples, compacta, rápida e poderosa.

A linguagem FORTH é muito flexível. Com ela, ao contrário das outras linguagens, nós podemos definir novos comandos a partir dos já existentes, ou então diretamente em linguagem de máquina. Neste artigo, desenvolveremos novos comandos extremamente úteis, inclusive um que permite salvar o compilador FORTH em disco. Mas, atenção!!! Todas as palavras aqui propostas devem ser criadas usando base 16. Para passar para a base 16, utilize o comando HEX definido no artigo publicado no número 22.

A possibilidade de incluir novos comandos, entretanto, cria uma dificuldade: é necessário lembrar quais as palavras que já foram definidas em nosso FORTH. Por isso, alguns FORTHS trazem em seu vocabulário a palavra VLIST ou outra similar, a qual serve para listar o vocabulário. (É possível implementar VLIST neste compilador, e é o que faremos mais adiante).

Uma palavra FORTH, na versão que foi apresentada em MICRO SISTEMAS, fica armazenada na memória da seguinte forma:

- 1 byte com o número de letras da palavra;
- a palavra;
- 2 bytes com um *link* com a palavra anterior;
- 2 bytes com o endereço do código de máquina associado a ela;

- parâmetros (dados ou endereços de outras palavras).
Vamos usar como exemplo a palavra END,:

ENDEREÇO	CONTEÚDO	COMENTÁRIOS
6177	04	Numero de letras da palavra
6178	"END,"	A palavra
617C	65 61	Link com a palavra anterior (F_TOKEN)
617E	0A 5E	Endereço da rotina em linguagem de máquina associada com esta palavra.
6180	EB 65	Endereço da palavra ","
6182	2F 66	Endereço da palavra "HERE"
6184	EB 62	Endereço da palavra "-"
6186	03 68	Endereço da palavra "C,"
6188	F2 50	Endereço da rotina de retorno.

Note que todos os endereços estão na ordem LSB-MSB (byte menos significativo — byte mais significativo). A primeira palavra tem *link* = 0 para indicar o início do vocabulário. Como podemos ver, para listar o vocabulário FORTH nós devemos começar pelo fim e terminar no início.

Assim sendo, para listar o dicionário bastaria criar uma palavra que tirasse o endereço da pilha, listasse a palavra associada a ele e em seguida colocasse na pilha o *link* desta palavra com a anterior, repetindo a execução desta palavra até que o *link* encontrado fosse zero. Porém, não basta somente isso...

Nesta versão do FORTH existem dois vocabulários. Além do vocabulário comum, existe um outro, que se chama vocabulário de compilação. O vocabulário de compilação tem a mesma estrutura do vocabulário comum, com apenas uma diferença: o endereço de sua última palavra é dado por COMPILER@, enquanto que o endereço da última palavra do vocabulário comum é obtido pela palavra ENTRY.

VAMOS À IMPLEMENTAÇÃO

A palavra que lista uma única palavra será escrita em linguagem de máquina para aproveitar uma sub-rotina que está incluída no compilador. Esta rotina começa em 5E47H e termina em 5E50H, e sua função é listar uma *string* cujo comprimento está no registrador E e cujo endereço inicial está no registrador HL. No retorno, HL aponta para a última letra da *string*. Eis a nossa palavra escrita em linguagem de máquina:

E1	POP	HL	; tira endereço da pilha
5E	LD	E,(HL)	; le número de letras da
			; palavra
23	INC	HL	; aponta para o início
			; da palavra
CD 47 5E	CALL	5E47	; lista a palavra
23	INC	HL	; aponta para o link
5E	LD	E,(HL)	; carrega o link no par
23	INC	HL	; de registradores
56	LD	D,(HL)	; DE.
05	PUSH	DE	; coloca link na pilha
C3 FE 50	JP	NEXT	; retorna

Para implementar esta mesma palavra em FORTH, digite:

```
CREATE LISTA-PALAVRA E1 C, 5E C, 23 C, CD C, 5E47 ,
23 C, 5E C, 23 C, 56 C, 05 C, NEXT
```

Muita atenção, pois pelo menos um espaço deve ser colocado entre cada símbolo da linguagem (números ou palavras). A palavra NEXT encarrega-se de colocar os bytes C3 FE 5D.

Agora precisamos implementar uma palavra que liste um dos vocabulários completamente. Ela irá buscar na pilha o endereço da última palavra do vocabulário. Para implementá-la, digite:

```
: LISTA-VOC
  BEGIN
    LISTA-PALAVRA
    ASpace DUP EMIT EMIT
    DUP 0=
  UNTIL
  OK
```

Observe que esta palavra usa as palavras BEGIN e UNTIL, que foram definidas em MICRO SISTEMAS número 22. Você deverá incluí-las no vocabulário antes de definir a palavra LISTA-VOC.

Finalmente, vamos implementar a palavra VLIST, a qual listará os dois vocabulários:

```
: VLIST
  CR ENTRY LISTA-VOC
  CR COMPILER @ LISTA-VOC
  CR
```

PALAVRAS ÚTEIS EM JOGOS

FORTH é uma linguagem muito boa para a criação de jogos. Existem, inclusive, algumas palavras que são imprescindíveis para isto, como por exemplo: INKEY (varre o teclado sem esperar pelo pressionamento de uma tecla); P@ (lê um byte de uma porta); P! (transmite um byte para uma porta); SET (seta um ponto na tela); RESET (resseta um ponto na tela); POINT (testa se um ponto está setado ou não). Vamos, então, implementar todas estas palavras.

• INKEY – Faz uma varredura do teclado. Se alguma tecla estiver sendo pressionada, o código desta será colocado na pilha; caso contrário, um zero será colocado na pilha:

CD 2B 00	CALL	002B	; varre teclado
4F	LD	L,A	; carrega L com o código
			; da tecla pressionada
26 00	LD	H,0	; e H com 0
E5	PUSH	HL	; coloca na pilha
C3 FE 50	JP	NEXT	; retorna

CARO CONSUMIDOR

*Mais barato impossível.
E o cliente é quem
faz o financiamento*

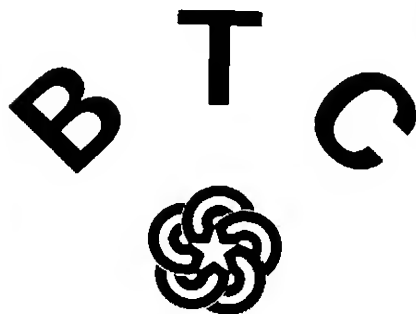
TK 2000	Cr\$ 849.850,00
TK 85	Cr\$ 369.850,00
Dactron	Cr\$ 1.299.000,00
CP 500 c/2 Drives	Cr\$ 3.900.000,00
Intellivision	Cr\$ 240.000,00
Mônica	Cr\$ 1.390.000,00
Drives	Cr\$ 649.850,00
Diskettes	Cr\$ 6.300,00
Jogos p/TK 2000 - a partir de	Cr\$ 9.000,00
Jogos p/Intellivision - a partir de	Cr\$ 25.000,00
Placas CPM Probus	Cr\$ 200.000,00
Programas Aplicativos	Cr\$ 20.000,00
Fitas p/Video Cassete	Cr\$ 17.500,00
Cabo p/Impressora	Cr\$ 90.000,00
Mesas p/Computador	varia de acordo
Mesas p/Impressora	com o modelo

Temos diversos outros produtos com o melhor preço do mercado. Os preços são da ORTN do mês de Abril, podendo aumentar ou diminuir de acordo com os produtos que estarão em promoção da semana.

Temos Curso de Basic para adultos e crianças com aulas teóricas e práticas com Micros.

Diversas turmas e horários.

Venha fazer-nos uma visita.



BRASILTRADE CENTER

Av. Epitácio Pessoa, 280 (Esq. com Visc. de Pirajá) (Aberto diariamente até às 22:00h)

CEP 22471, tels.: (021) 259-1299, 259-1499 e 259-1542.

Assembléia, 10 - loja 112 — Centro Empresarial Cândido Mendes.

Em FORTH,

CREATE INKEY CD C, 2B, 6F C, 26 C, 0 C, E5 C, NEXT

- P@ — Lê um byte da porta cujo número está na pilha; o byte será colocado na pilha:

```
D9      EXX      BC      ; numero da porta
C1      POP      E,(C)    ; le byte
ED 58   IN       D,0      ;
16 00   LD       DE       ; coloca byte na pilha
D5      PUSH     DE
D9      EXX
C3 FE 5D JP      NEXT
```

Em FORTH,

CREATE P@ D9 C, C1 C, ED C, 58 C, 16 C, 0 C, D5 C, D9 C, NEXT

- P! — Transmite um byte para uma porta. O número da porta está no TOS e o byte em 2OS. Esta palavra é particularmente útil na criação de efeitos sonoros:

```
D9      EXX      BC      ; porta
C1      POP      DE       ; byte
D1      POP      (C),E    ; transmite
ED 59   OUT
D9      EXX
C3 FE 5D JP      NEXT
```

Em FORTH,

CREATE P! D9 C, C1 C, D1 C, ED C, 59 C, D9 C, NEXT

Agora, antes de passarmos para as palavras SET, RESET e POINT, vamos criar uma palavra chamada GRAF em Assembler, que atuará como uma sub-rotina das outras três palavras. Esta palavra foi baseada em uma rotina do interpretador BASIC:

```
D9      EXX      DE      ; salva registradores
D1      POP      A,E      ; tira coordenada x
7B      LD       A,E      ; e coloca em A
FE 30   CP       30      ; >= 48 (decimal)?
38 18   ---JR    C,HERE+18 ; nao, salta
21 ?? ?? ! LO <--- HL,HERE+0B ; sim, imprime mensagem
1E 0C   ! LO      E,0C    ; de erro
CD 47 5E ! CALL   SE47
D9      ! EXX          ; recupera registradores
C3 FE 5D ! JP      NEXT  ; e retorna.
46 4F 52 41 ! DEFM 'FORA DA TELA'
20 44 41 20 !
54 45 4C 41 !
16 FF   -->LD     D,FF
14      -->INC    D      ; D=quociente da divisao
06 03   ! SUB     03     ; do registrador A por tres
30 FB   ---JR    NC,HERE-5 ;
C6 03   ADD      03     ; C=resto
4F      LD       C,A     ; retira coordenada y
E1      POP      HL      ; e coloca em A
7D      LD       A,L     ; >= 128 (decimal)?
FE 80   CP       80     ; sim, volta e imprime
30 DB   JR      NC,HERE-28 ; mensagem de erro
87      ADD      A       ; nao, multipl. por 2
5F      LD       E,A     ; e coloca em E
06 02   LD       B,02
7A      -->LD     A,D     ; move D e E duas vezes
1F      ! RRA      ; para a direita assim o
57      ! LD       D,A    ; bit 1 de E vai para a
7B      ! LD       A,E    ; flag CARRY e o par DE
1F      ! RRA      ; aponta para o carac.
5F      ! LD       E,A    ; grafico que contem o
10 FB   ---DJNZ  HERE-08 ; ponto.
79      LD       A,C     ; Calcula a posicao do
8F      ADC      A       ; ponto dentro do carac.
3C      INC      A       ; grafico usando a for-
47      LD       B,A     ; mula (2*C)+1+CARRY
AF      XOR      A       ; coloca a posicao de
37      SCF      ; ponto em B
8F      -->ADC    A       ; cria uma mascara com
!       !             ; o registrador A setan-
!       !             ; do o bit apontado pelo
!       !             ; valor de B e resetando
```

```
10 FD   ---DJNZ  HERE-05 ; os outros bits.
4F      LD       C,A     ; salva a mascara em C
7A      LD       A,D     ; DE aponta para o cara-
F6 3C   OR       3C     ; cter da tela onde es-
57      LD       D,A     ; ta' o ponto.
1A      LD       A,(DE)  ; carrega o carac. em A
87      OR       A       ; e' caractere grafico?
FA ?? ?? ---JP    M,HERE+4 ; sim, salta
3E 80   ! LD     A,80    ; nao, mas agora e'
47      -->LD    B,A     ; salva caractere em B
F1      POP      AF      ; tira flag da pilha
B7      OR       A       ; testa
7B      LD       A,B     ; recupera caractere
28 11   JR      Z,HERE+11 ; flag=0? sim, pula
12      LD       (DE),A  ; recoloca car. na tela
FA ?? ?? ---JP    M,HERE+0C ; flag=80? sim, pula
79      ! LD     A,C     ; coloca mascara em A
2F      ! CPL      ; complementa
4F      ! LD     C,A     ; recoloca em C
1A      ! LD     A,(DE)  ; coloca caractere gra-
!       !             ; fico em A
A1      ! AND     C       ; reseta o ponto
12      ! LO <--- (DE),A ; recoloca o caractere
D9      ! ---EXX      ; recupera registradores
C3 FE 5D ! ! JP      NEXT ; e retorna
B1      ! ---OR     C     ; seta o ponto
18 FB   ! JR      HERE-08 ; volta
A1      ! AND     C       ; testa o ponto.
21 00 00 ! LD     HL,0000 ; flag de retorno da
!       !             ; palavra POINT
28 01   ! ---JR    Z,HERE+1 ; ponto esta' resetado?
2C      ! ! INC     L      ; nao, L=01
E5      ! -->PUSH  HL      ; coloca na pilha
18 EF   ---JR    HERE-11 ; volta
```

Em FORTH,

```
CREATE GRAF D9 C, D1 C, 7B C, FE C, 30 C, 38 C, 18 C,
21 C, HERE B +, 1E C, 0C C, CD C, SE47, D9 C, C3 C, 5D FE,
46 C, 4F C, 52 C, 41 C, 20 C, 44 C, 41 C, 20 C,
54 C, 45 C, 4C C, 41 C, 16 C, FF C, 14 C, 06 C, 03 C, 30 C,
FB C, C6 C, 03 C, 4F C, E1 C, 7D C, FE C, 80 C, 30 C, DB C,
87 C, 5F C, 06 C, 02 C, 7A C, 1F C, 57 C, 7B C, 1F C, 5F C,
10 C, FB C, 79 C, 8F C, 3C C, 47 C, AF C, 37 C, 8F C, 10 C,
FD C, 4F C, 7A C, F6 C, 3C C, 57 C, 1A C, 87 C, FA C,
HERE 4 +, 3E C, 80 C, 47 C, F1 C, B7 C, 7B C, 28 C, 11 C, 12 C,
FA C, HERE C +, 79 C, 2F C, 4F C, 1A C, A1 C, 12 C, D9 C, C3 C,
5D FE, B1 C, 18 C, FB C, A1 C, 21 C, 0, 28 C, 01 C, 2C C,
ES C, 18 C, EF C,
```

Como esta rotina é longa, eu aconselho o leitor a digitá-la aos poucos, no máximo 20 bytes de cada vez à medida que a confere, pois a possibilidade de erros é maior.

- SET — Seta um ponto na tela cujas coordenadas X e Y são respectivamente o 2OS e o TOS:

: SET 8000 ROT ROT GRAF ;

- RESET — Resseta um ponto e é similar à anterior quanto às coordenadas do ponto:

: RESET 100 ROT ROT GRAF ;

- POINT — Testa se um ponto está setado ou não. Se estiver, devolve 1 na pilha; caso contrário, devolve 0:

: POINT 0 ROT ROT GRAF ;

SALVANDO EM DISCO

Finalmente, vamos apresentar uma palavra que permite executar comandos de disco em FORTH. Esta palavra poderá ser usada com o sistema operacional NEWDOS/80 ou DIG-DOS100. Ela usa duas rotinas do sistema operacional. A primeira é a rotina que executa um comando do NEWDOS e a outra é a que mostra uma mensagem de erro.

Inicialmente, vamos implementar uma rotina em linguagem de máquina que apanha o endereço do comando na pilha e o executa, imprimindo eventuais mensagens de erro que possam ocorrer durante a execução do comando:

E1	POP	HL	; retira endereço do co- mando.
CD 19 44	CALL	4419	; executa
28 05	JR	Z,05	; salta se não houve erro
F6 80	OR	80	; seta bit mais signifi- cativo de A
CD 09 44	CALL	4409	; imprime mens. de erro
C3 FE 50	JP	NEXT	; retorna

Implementado em FORTH,

CREATE CMD-DOS E1 C, CD C, 4419 , 28 C, 05 C, F6 C, 80 C,
CD C, 4409 , NEXT

Feito isso, vamos então implementar uma palavra que pre-
para o comando para ser usado por CMD-DOS:

```

: CMD"
  22 TOKEN
  HERE DUP DUP
  CR + 1+ 0D SWAP C!
  1+ CR CMD-DOS
;

```

Note que esta palavra deve ser criada na base hexadecimal.
Para utilizá-la, digite CMD" comando-dos". Por exemplo: pa-
ra obter o diretório, digite CMD" DIR" (o espaço colocado de-
pois das aspas é obrigatório).

Nós podemos usar este comando para salvar o compilador
em disco. Para fazer isso, use a seguinte sequência:

- 1 - carregue o NEWDOS;
- 2 - passe para o BASIC e carregue o FORTH da fita usando o
comando SYSTEM;
- 3 - inclua as palavras CMD-DOS e CMD";
- 4 - digite DECIMAL HERE. (isto imprimirá o endereço do
final do compilador);
- 5 - digite o comando CMD" DUMP FORTH/CMD,24001,
nnnnn,24145", onde nnnnn é o número devolvido pelo item
4 da sequência.

Isto salvará o compilador em disco. Use esta sequência a
partir do item 4 sempre que quiser salvar as modificações feitas
no compilador. Os itens 4 e 5 deverão ser usados sem base
decimal. Feito isso, para carregar o FORTH em disco bastará
digitar o seu nome e o sistema operacional se encarregará do
resto. Um cuidado apenas: nem todos os comandos de disco
podem ser usados pelo FORTH, uma vez que alguns deles
ocupam a área de memória onde o FORTH se encontra.

Para terminar, um último lembrete: se você pretende criar
palavras em Assembler, atenção para não destruir o conteúdo
dos registradores BC, IX e IY, pois eles têm funções especí-
ficas dentro do compilador.

BIBLIOGRAFIA

- HOGAN, Thom, *Discover FORTH*, Osborn/McGraw-Hill.
- FARVOUR, James, *Microsoft BASIC Decoded & Other
Mysteries*, IJG.
- GRATZER, George A. & Thomas G, *Fast BASIC Beyond
TRS-80 BASIC*, John Wiley & Sons, INC.
- HUTTY, Roger, *Z80 Assembly Language Programming
for Students*, The MacMillan Press LTD.



Ivan Camilo da Cruz domina várias linguagens de programação, dentre
elas o BASIC, Pascal, FORTRAN, ALGOL, COBOL, PL/I e Assembler,
tendo experiência de programação em computadores grandes, minis e
micros (CP-500, DGT-100 etc.). Atualmente faz curso de Física na
UFRJ e trabalha no Instituto de Química da mesma universidade, onde
desenvolve um sistema contábil e vários outros programas administra-
tivos.

MICRO NEWS

Microcomputadores com crédito direto ou leasing

TK-2000	Cr\$ 849.850,
TK-85	Cr\$ 369.850,
CP-200	Cr\$ 369.000,
DGT-1000	Cr\$ 1.498.000,
CP-500	Cr\$ 1.990.000,
APPLE II PLUS	Cr\$ 1.590.000,

GRÁTIS!
UM CURSO DE BASIC
ENTREGA RÁPIDA EM TODO BRASIL

Preços sujeitos a alteração

Aplicativos: controle de estoque; contabilidade; folha de
pagamento; contas a receber pagar; mala direta; cadastro
de clientes e desenvolvimento de software para cada neces-
sidade.

Temos toda linha de periféricos e suprimentos para acom-
panhar o crescimento de sua empresa.

VISITE-NOS OU SOLICITE UM REPRESENTANTE

MICRONEWS COMÉRCIO E SERVIÇOS LTDA.
R. Assembléia 10 Gr. 3317 - Ed. Centro Cândido Mendes
Tel.: (021) 252-9420 - CEP 20011/RJ.

SEJA NOSSO COLABORADOR E ENTRE PARA A HISTÓRIA... ...de MICRO SISTEMAS, é claro!

Como? É simples:

- Se você desenvolveu um
programa interessante, escreva
um texto datilografado expli-
cando qual a sua utilização e
junte exemplos de aplicação.
Atenção: é imprescindível que
a listagem seja datilografada.
Se for possível mande tam-
bém, junto com a listagem
datilografada, uma fita cassete
ou disquete com o programa.
- Se for artigo, use também
a máquina de escrever e, caso
haja necessidade de desenhos e
ilustrações, detalhe-os o má-
ximo possível.

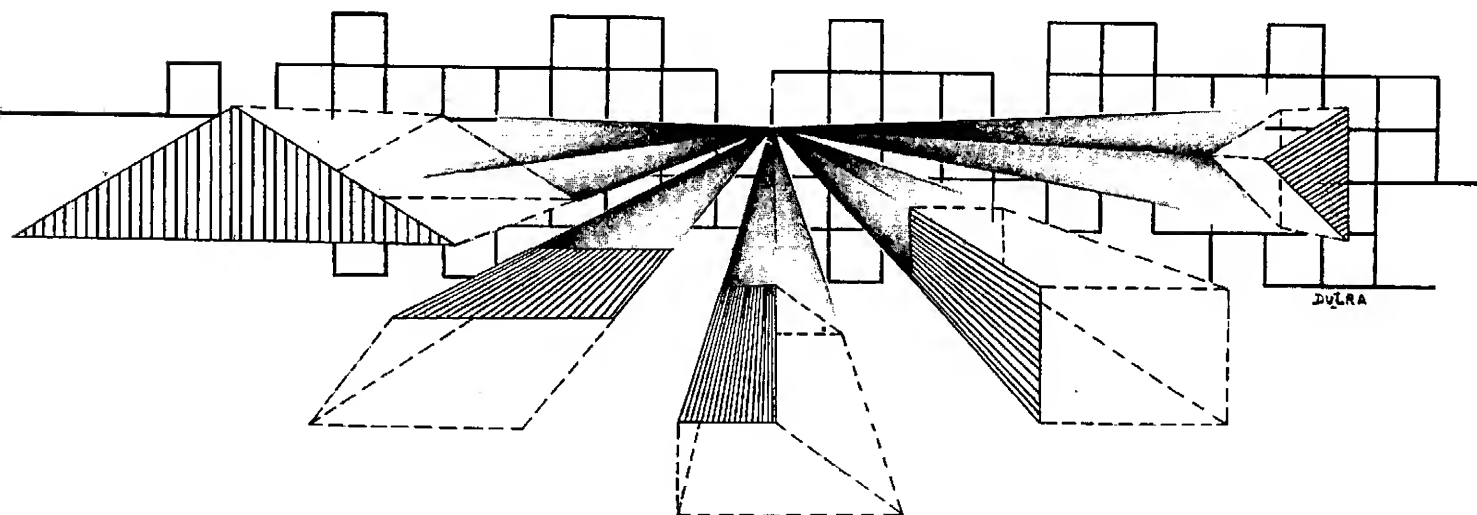
- Leia atentamente sua co-
laboração para micros ou cal-
culadora, veja se não falta ne-
nhuma informação (qual equi-
pamento, em que configuração
etc.) e remeta-a, em duas vias,
para a equipe de MICRO SIS-
TEMAS analisar.
- Se aprovado, seu artigo ou
programa será publicado em
data a ser confirmada na épo-
ca. Todo material veiculado é
remunerado. Caso não seja
aprovado, você receberá seu
material de volta.
- Não se esqueça de mandar
um breve currículo, seu nome,
telefone e endereço completo.

Micro Sistemas

Envie para REDAÇÃO/MICRO SISTEMAS, Rio ou São
Paulo: Av. Presidente Wilson, 165/grupo 1210, Centro, CEP
20030, Rio de Janeiro, RJ; Rua Oliveira Dias, 153, Jardim
Paulista, CEP 01433, São Paulo, SP.

Escolha a figura, dê as medidas e espere... Com este programa, seu micro TRS-80 I ou III calculará a área para você

Em geometria, a área é do micro



Heber Jorge da Silva

Cálculos de áreas é um programa que roda em qualquer micro compatível com o TRS-80 modelos I e III com 16 Kb de RAM e destina-se a calcular a área de 20 figuras

geométricas diferentes, das mais simples às mais complexas.

Logo em seguida à apresentação do programa, o usuário receberá na tela um menu com as opções correspondentes às

20 figuras a serem calculadas (figura 1). Escolhendo uma, o computador plotará no vídeo a figura requisitada, solicitando ao usuário entrar com as medidas necessárias aos cálculos. Ao informá-las, deve-se ter o cuidado de usar sempre a mesma unidade de medida, pois o programa não faz conversões.

Observações: as impressoras costumam imprimir de forma diferente o caráter de exponenciação. Os mais utilizados são: δ , Δ , \square e \uparrow . Em nossa listagem esta operação equivale ao caráter \square . Portanto, durante a digitação, substitua-o adequadamente.

```

**          ** MENU **
**          CALCULOS DE AREAS          **

1 - Quadrado
2 - Retangulo
3 - Paralelogramo
4 - Triangulo Retangulo
5 - Triangulo Acutangulo
6 - Triangulo Obtusangulo
7 - Trapezio
8 - Trapezoide
9 - Triangulo Equilatero
10 - Pentagono

11 - Hexagono
12 - Heptagono
13 - Octogono
14 - Circulo
15 - Coroa Circular
16 - Setor Circular
17 - Seguimento Circular
18 - Setor Coroa Circular
19 - Elipse
20 - Parabola

S E L E C I O N E ? M
    
```

Figura 1

Formado em Administração de Empresas pela UDF, Heber Jorge da Silva trabalha atualmente na Telebrasil, onde exerce atividades ligadas à programação de microcomputadores.

Cálculos de Áreas

```

5  *****
10 *****
15 ***      CALCULO DE AREAS      ***
16 ***      POR  HEBER JORGE DA SILVA      ***
16 ***      BRASILIA - DF      03/83      ***
20 *****
25 REM ***** MENU PRINCIPAL
30 CLS:PRINT@454,CHR$(23);***      CALCULOS DE A
REAS      ***:FORD=1T01500:NEXT
40 CLS:PRINTTAB(27)*** MENU ***:PRINTTAB(14)"
**      CALCULOS DE AREAS      ***:PRINT:PR
INT
50 PRINTTAB(10)"1 - Quadrado      1
1 - Hexagono":PRINTTAB(10)"2 - Retangulo
12 - Heptagono":PRINTTAB(10)"3 - Pa
ralelogramo      13 - Octogono"
60 PRINTTAB(10)"4 - Triangulo Retangulo      1
4 - Circulo":PRINTTAB(10)"5 - Triangulo Acuta
ngulo      15 - Coroa Circular":PRINTTAB(10)"6
- Triangulo Obtusangulo      16 - Setor Circular
"
70 PRINTTAB(10)"7 - Trapezio      1
7 - Segmento Circular"
80 PRINTTAB(10)"8 - Trapezoide      1
8 - Setor Coroa Circular":PRINTTAB(10)"9 - Tr
iangulo Equilatero      19 - Elipse":PRINTTAB(9
)"10 - Pentagono      20 - Parabola"
90 PRINT:INPUT"      S E L E
C I O N E ";A:IFA(10A)20THEN90
100 GOSUB1440:ONAGOTO110,290,450,520,610,630,
650,670,700,720,730,740,750,760,770,810,830,8
60,890,920
110 GOSUB950:GOSUB960
120 ONGGOTO130,190,240
130 CLS:PRINT@64,"CALCULO DA AREA DO QUADRADO
":GOSUB950
140 FORX=1T043:FORY=1T029:SET(X,Y)=NEXTY,X
150 PRINT@704,"":INPUT"1 - atraves da diagona
1 2 - atraves do lado";A:IFA(10A)2THEN150
160 ONAGOTO170,180
170 GOSUB1510:GOSUB1400:V=.5*(DE2):GOTO1600
180 GOSUB1510:GOSUB990:V=LC2:GOTO1600
190 CLS:PRINT@64,"CALCULO DOS LADOS DO QUADRA
DO":GOSUB950
200 PRINT@704,"":INPUT"1 - Atraves da area
2 - Atraves da diagonal";L:IFL(10RL)2THEN200
210 ONLGOTO220,230
220 GOSUB1510:GOSUB1390:V=SQR(A):GOTO1600
230 GOSUB1510:GOSUB1400:V=D*.7071:GOTO1600
240 CLS:PRINT@64,"CALCULO DA DIAGONAL DO QUAD
RADO":GOSUB950
250 PRINT@704,"":INPUT"1 - Atraves da area
2 - Atraves do lado";D:IFD(10RD)2THEN250
260 ONDGGOTO270,280
270 GOSUB1510:GOSUB1390:V=SQR(A)*1.414:GOTO16
00
280 GOSUB1510:GOSUB990:V=1.414*L:GOTO1600
290 GOSUB1000:GOSUB960
300 ONGGOTO310,400,430
310 CLS:PRINT@128,"CALCULO DA AREA DO RETANGU
LO":GOSUB1000
320 FORX=1T059:FORY=17T029:SET(X,Y)=NEXTY,X
330 PRINT@704,"":INPUT"ATRAVES DE: 1 - diagon
al+lado maior 2 - diagonal+lado menor
3 - lado maior+lado menor"
;A:IFA(10A)3THEN330
340 ONAGOTO350,370,390
350 GOSUB1510:GOSUB1520:GOSUB1400:GOSUB1450:I
FL2=>DTHENGOSUB970:GOTO330
360 V=L2*SQR((DE2)-(L2E2)):GOTO1600
370 GOSUB1510:GOSUB1520:GOSUB1400:GOSUB1440:I
FL1=>DTHENGOSUB970:GOTO330
380 V=L1*SQR((DE2)-(L1E2)):GOTO1600
390 GOSUB1510:GOSUB1520:GOSUB1410:GOSUB1420:V
=M*N:GOTO1600
400 CLS:PRINT@128,"CALCULO DOS LADOS DO RETAN
GULO":GOSUB1000
410 PRINT@704,"":GOSUB1400:INPUT"Entre com um
dos lados";O:IFD(<0THENGOSUB970:GOTO410
420 V=SQR((DE2)-(OE2)):GOTO1600
430 CLS:PRINT@128,"CALCULO DA DIAGONAL DO RET
ANGULO":GOSUB1000
440 PRINT@704,"":GOSUB1410:GOSUB1420:V=SQR((M
E2)+(NE2)):GOTO1600
450 GOSUB1010:GOSUB1020
460 ONGGOTO470,490
470 CLS:PRINT@128,"CALCULO DA AREA DO PARALEL
OGRAMO":GOSUB1010:A=1:B=60:Y=17:FORD=1T013:GO
SUB1670:A=A+1:B=B+1:Y=Y+1:NEXT
480 PRINT@704,"":INPUT"Entre com os valores d
os lados";M,N:V=M*N:GOTO1600
490 CLS:PRINT@128,"CALCULO DOS LADOS DO PARAL
ELOGRAMO":GOSUB1010
500 PRINT@704,"":GOSUB1390:GOSUB1410:IFA(<MTH
ENGOSUB970:GOTO500
510 V=A/M:GOTO1600
520 GOSUB1030:PRINT@704,"O QUE QUER CALCULAR
?":PRINT:PRINT"1 - Area      2 - Cateto      3
- Hipotenusa":PRINT:INPUT"      ESCOLHA..."
;Q:IFQ(10RQ)3THEN520
530 ONGGOTO540,560,590
540 CLS:PRINT@0,"CALCULO DA AREA DO TRIANGULO
RETANGULO":GOSUB1030:Y1=5:FORX=1T040:FORY=Y
1T030:SET(X,Y)=NEXT:Y1=Y1+1:NEXT
550 PRINT@704,"":INPUT"Entre com os dois cate
tos";M,N:V=(M*N)/2:GOTO1600
560 CLS:PRINT@0,"CALCULO DE CATETOS DO TRIANG
ULO RETANGULO":GOSUB1030
570 PRINT@704,"":INPUT"Entre com a hipotenusa
";H:INPUT"Entre com um dos catetos";O:IFH(<0T
HENGOSUB970:GOTO570
580 V=SQR((HE2)-(OE2)):GOTO1600
590 CLS:PRINT@0,"CALCULO DA HIPOTENUSA DO TRI
ANGULO RETANGULO":GOSUB1030
600 PRINT@704,"":INPUT"Entre com os dois cate
tos";M,N:V=SQR((ME2)+(NE2)):GOTO1600
610 PRINT@128,"CALCULO DA AREA DO TRIANGULO A
CUTANGULO":GOSUB1040
620 GOSUB980:V=B*H/2:GOTO1600
630 PRINT@128,"CALCULO DA AREA DO TRIANGULO O
BTUSANGULO":GOSUB1050

```

SUPRIMENTO É COISA SÉRIA

• Mantenha o seu computador bem alimentado adquirindo produtos de qualidade consagrada.

Discos Magnéticos: 5 Mb, 16 Mb, 80 Mb etc.
DISKETTES: 5 1/4 e 8" - marca VERBATIM
ETIQUETAS PIMACO - PIMATAB

• Fita Magnética: 600, 1200 e 2400 Pés
• Fita CARBONITAS p/Impressoras: Globus M 200 — B 300/600
• Fita p/Impressoras: Elebra, Elgin, Epson, Digilab, Diabla, etc.
• Cartucho Cobra 400
• Pastas e Formulários Contínuos.

AV. PRESIDENTE VARGAS N° 482 GR 207 TEL (021) 253-1120 E 263-5876

CENTRALDATA
Com. e Representações Ltda.

```

640 GOSUB980:V=B*H/2:GOTO1600
650 PRINT@128,"CALCULO DA AREA DO TRAPEZIO":G
OSUB1060
660 PRINT@704,"":GOSUB1430:GOSUB1450:GOSUB144
0:V=H*(L2+L1)/2:GOTO1600
670 PRINT@128,"CALCULO DA AREA DO TRAPEZOIDE"
:GOSUB1070
680 PRINT@704,"":INPUT"Entre com a altura me
nor (h)":H1=INPUT"Entre com a altura maior (H
)":H2=INPUT"Entre com a base (A)":L=INPUT"Ent
re com a base (B)":B=INPUT"Entre com a base (
C)":C
690 V=(H1+H2)*L+B*H1+C*H2/2:GOSUB1510:GOSUB
1520:GOSUB1540:PRINT@640,STRING$(30,31):GOSUB
1560:GOSUB1570:GOTO40
700 PRINT@64,"CALCULO DA AREA DO TRIANGULO EQ
UILATERO":GOSUB1090
710 GOSUB990:V=.433*(L2):GOTO1600
720 PRINT@64,"CALCULO DA AREA DO PENTAGONO":G
OSUB1100:GOSUB990:V=1.720*(L2):GOTO1600
730 PRINT@64,"CALCULO DA AREA DO HEXAGONO":GOS
UB1110:GOSUB990:V=2.598*(L2):GOTO1600
740 PRINT@0,"CALCULO DA AREA DO HEPTAGONO":GOS
UB1120:PRINT@704,"":GOSUB990:V=3.634*(L2):G
OTO1600
750 PRINT@0,"CALCULO DA AREA DO OCTOGONO":GOS
UB1130:GOSUB990:V=4.828*(L2):GOTO1600
760 PRINT@0,"CALCULO DA AREA DO CIRCULO":GOSU
B1150:FORX=32TO57STEP2:RESET(X,17):NEXT:POKEP
+350,82:PRINT@704,"":GOSUB1650:V=3.14159*(R2
):GOTO1600
770 PRINT@0,"CALCULO DA AREA DA COROA CIRCULA
R":GOSUB1150:GOSUB1180:POKEP+403,82:POKEP+397
,114
780 PRINT@704,"":GOSUB1460:GOSUB1470
790 IFR1)=R2THENGOSUB970:GOTO780
800 V=3.14159*(R2L2)-(R1L2):GOTO1600
810 PRINT@220,"CALCULO DA AREA DO SETOR CIRC
ULAR":GOSUB1210:POKEP+212,114:RESET(30,13):RES
ET(30,21):X=29:FORY=13TO16:RESET(X,Y):X=X-1:N
EXT:X=29:FORY=21TO18STEP-1:RESET(X,Y):X=X-1:N
EXT
820 RESET(26,17):PRINT@333,"alfa":GOSUB1480:
GOSUB1650:V=.008727*Z*(R2):GOTO1600
830 PRINT@218,"CALCULO DA AREA DO SEGMENTO CI
RCULAR":GOSUB1250:POKEP+258,65:POKEP+322,76:P
OKEP+386,70:POKEP+450,65:POKEP+342,67:FORX=26
TO45STEP4:SET(X,6):SET(X,28):NEXT:FORX=9TO23S
TEP2:RESET(X,17):NEXT:POKEP+328,104
840 FORY=6TO28STEP2:SET(42,Y):NEXT:SET(7,10):
SET(7,25):SET(10,7):SET(10,28):SET(14,4):SET(
14,31):RESET(25,8):RESET(27,10):RESET(29,12):
RESET(31,14):RESET(33,16):POKEP+208,114
850 GOSUB1480:GOSUB1650:INPUT"Entre com o val
or do lado (C)":C:GOSUB1430:V=.5*(R*(.01745*R
*Z)-C*(R-H)):GOTO1600
860 PRINT@18,"CALCULO DA AREA DO SETOR DE COR
OA CIRCULAR":POKEP+275,114:POKEP+469,82:GOSUB
1280:PRINT@333,"alfa":
870 GOSUB1480:GOSUB1460:GOSUB1470:IFR1)=R2THE
NGOSUB970:GOTO870
880 V=Z*3.14159/360*((R2L2)-(R1L2)):GOTO1600
890 PRINT@4,"CALCULO DA AREA DA ELIPSE":GOSUB
1340
900 GOSUB1490:GOSUB1500:IFX(=YTHENGOSUB970:GO
TO900
910 V=3.14159*X*Y:GOTO1600
920 PRINT@0,"CALCULO DA AREA DA PARABOLA":GOS
UB1360:GOSUB1370
930 GOSUB1470:GOSUB1500:IFX(=YTHENGOSUB970:GO
TO930
940 V=X*Y*2/3:GOTO1600
945 REM ***** DESENHA QUADRADO
950 FORX=0TO44:SET(X,10):SET(X,30):NEXT:FORY=
11TO30:SET(0,Y):SET(44,Y):NEXT:POKEP+138,76:P
OKEP+408,76:X=44:FORY=10TO29:SET(X,Y):X=X-2.3

```

```

: NEXT:POKEP+394,68:RETURN
960 PRINT@704,"O QUE QUER CALCULAR?":PRINT:P
RINT"1 - Area 2 - Lado 3 - Diagonal":PR
INT:INPUT" ESCOLHA...":G=IFQ(10RQ)3TH
EN960ELSERETURN
970 GOSUB1510:PRINT@704,"IMPOSSIVEL !!! FAVO
R REINFORMAR.":FORD=1TO1200:NEXT:RETURN
980 PRINT@704,"":INPUT"Entre com o valor da b
ase (B)":B=INPUT"Entre com o valor da altura
(h)":H:RETURN
990 PRINT@704,"":INPUT"Entre com o valor do l
ado":L:RETURN
995 REM ***** DESENHA RETANGULO
1000 FORX=0TO60:SET(X,16):SET(X,30):NEXT:FORY
=16TO30:SET(0,Y):SET(60,Y):NEXT:POKEP+270,76:
POKEP+481,108:POKEP+463,68:X=0:FORY=17TO29:SE
T(X,Y):X=X+4.7:NEXT:RETURN
1005 REM ***** DESENHA PARALELOGRAMO
1010 X=0:FORY=16TO30:SET(X,Y):SET(X+60,Y):X=X
+1:NEXT:FORX=0TO60:SET(X,16):SET(X+14,30):NEX
T:POKEP+271,76:POKEP+484,76:RETURN
1020 PRINT@704,"O QUE QUER CALCULAR?":PRINT:
PRINT"1 - Area 2 - Lados":INPUT"
ESCOLHA...":G=IFQ(10RQ)2THEN1020ELSERETU
RN
1025 REM ***** DESENHA TRIANGULO RETANGULO
1030 X=15:FORY=5TO30:SET(X,Y):X=X+1:NEXT:FORY
=5TO30:SET(15,Y):NEXT:FORX=15TO40:SET(X,30):N
EXT:POKEP+325,67:POKEP+335,72:POKEP+718,67:RE
TURN
1035 REM ***** DESENHA TRIANGULO ACUTANGULO
1040 A=20:B=22:FORY=12TO29:GOSUB1670:A=A-1:B=
B+3:NEXT:POKEP+457,104:POKEP+658,66:FORY=11TO
29STEP2:SET(21,Y):RESET(21,Y):NEXT:RETURN
1045 REM ***** DESENHA TRIANGULO OBTUSANGULO
1050 A=58:B=58:FORY=12TO29:GOSUB1670:A=A-3:B=
B-1:NEXT:POKEP+414,104:POKEP+654,66:FORY=12TO
29STEP2:SET(58,Y):NEXT:FORX=41TO61STEP4:SET(X
,29):NEXT:RETURN
1055 REM ***** DESENHA TRAPEZIO
1060 A=15:B=51:FORY=20TO29:GOSUB1670:A=A-1:B=
B+2:NEXT:FORY=19TO30STEP2:RESET(50,Y):NEXT:PO
KEP+336,67:POKEP+654,66:POKEP+538,104:RETURN
1065 REM ***** DESENHA TRAPEZOIDE
1070 A=50:B=60:FORY=19TO21:GOSUB1670:A=A-10:B
=B+1:NEXT:A=20:B=63:FORY=22TO29:GOSUB1670:A=A
-1:B=B+1:NEXT:POKEP+647,66:POKEP+523,104:POKE
P+660,65:POKEP+541,72:POKEP+673,67
1080 FORY=21TO30STEP2:RESET(20,Y):NEXT:FORY=1
9TO30STEP2:RESET(61,Y):NEXT:SET(20,31):SET(61
,31):RETURN
1085 REM ***** DESENHA TRIANGULO EQUILATERO
1090 A=36:B=36:FORY=8TO30:GOSUB1670:A=A-1.3:B
=B+1.3:NEXT:SET(37,9):POKEP+721,76:RETURN
1095 REM ***** DESENHA PENTAGONO
1100 A=36:B=36:FORY=8TO16:GOSUB1670:A=A-2.6:B
=B+2.6:NEXT:A=16:B=55:FORY=17TO27:GOSUB1670:A
=A-.62:B=B-.62:NEXT:POKEP+658,76:SET(54,18):R
ETURN
1105 REM ***** DESENHA HEXAGONO
1110 A=20:B=49:FORY=8TO18:GOSUB1670:A=A-1.15:
B=B+1.15:NEXT:RESET(49,8):A=20:B=49:FORY=29TO
19STEP-1:GOSUB1670:A=A-1.15:B=B+1.15:NEXT:RES
ET(49,29):POKEP+656,76:RETURN
1115 REM ***** DESENHA HEPTAGONO
1120 A=34:B=34:FORY=5TO11:GOSUB1670:A=A-4:B=B
+4:NEXT:A=10:B=58:FORY=12TO21:GOSUB1670:A=A-.
5:B=B+.5:NEXT:A=5:B=63:FORY=22TO32:GOSUB1670:
A=A+1.5:B=B-1.5:NEXT:POKEP+720,76:RETURN
1125 REM ***** DESENHA OCTOGONO
1130 A=34:B=34:FORY=5TO10:GOSUB1670:A=A-4.2:B
=B+4.2:NEXT:SET(33,5):A=12:B=55:FORY=11TO18:G
OSUB1670:A=A-1:B=B+1:NEXT:A=34:B=34:FORY=32TO
27STEP-1:GOSUB1670:A=A-4.2:B=B+4.2:NEXT
1140 SET(33,32):A=12:B=55:FORY=26TO19STEP-1:G
OSUB1670:A=A-1:B=B+1:NEXT:POKEP+650,76:RETURN

```

```

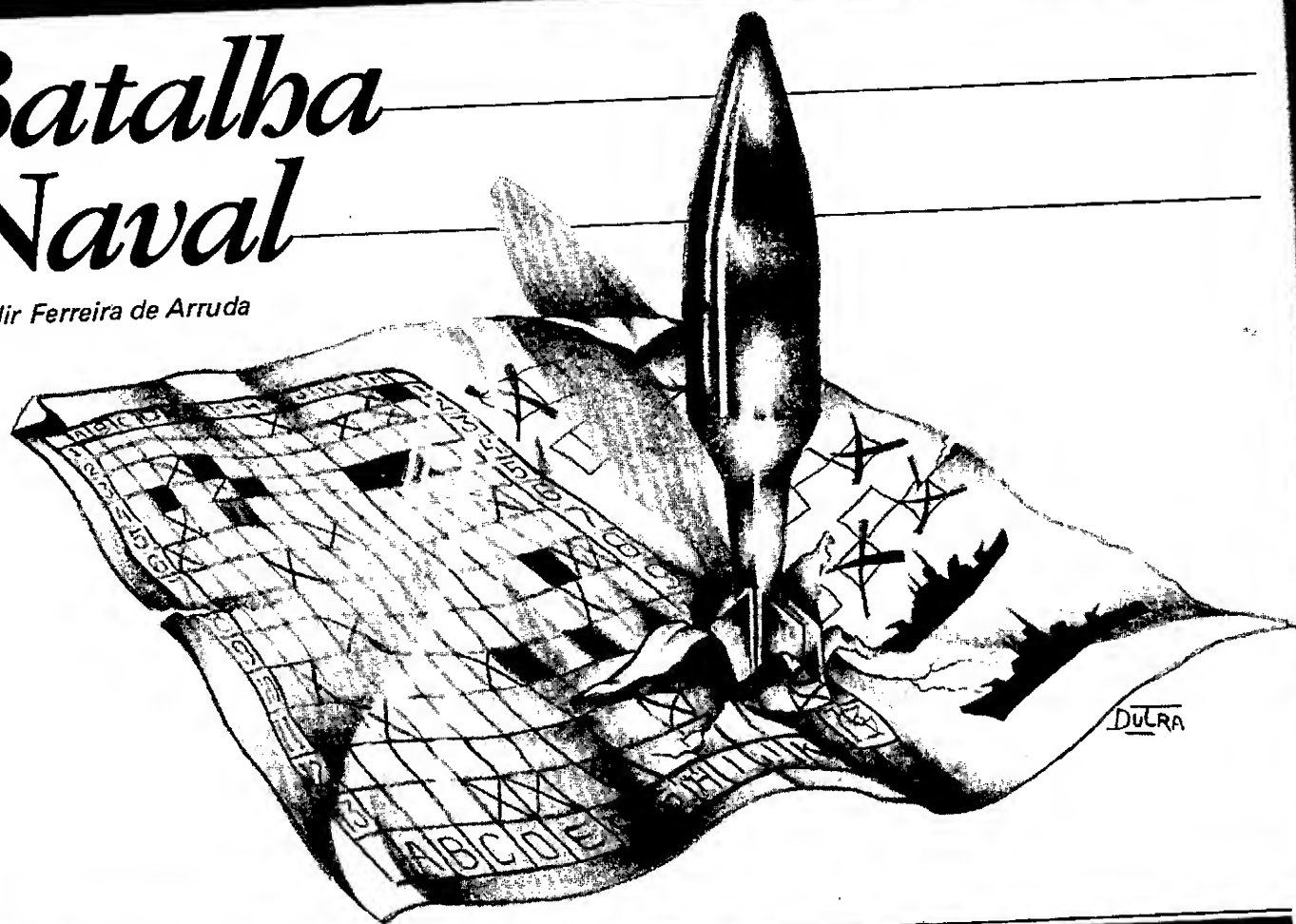
1145 REM ***** DESENHA CIRCULO
1150 A=24:B=38:FORX=5T07:GOSUB1670:A=A-4:B=B+
4:NEXT:A=14:B=48:FORX=8T011:GOSUB1670:A=A-2:B
=B+2:NEXT:FORX=7T055:SET(X,12):NEXT:FORX=13T0
14:GOSUB1170:NEXT:FORX=15T019:FORX=5T057:SET(
X,Y):NEXT:NEXT
1160 A=24:B=38:FORX=29T027STEP-1:GOSUB1670:A=
A-4:B=B+4:NEXT:A=14:B=48:FORX=26T023STEP-1:GO
SUB1670:A=A-2:B=B+2:NEXT:FORX=7T055:SET(X,22)
:NEXT:FORX=21T020STEP-1:GOSUB1170:NEXT:RETURN
1170 FORX=6T056:SET(X,Y):NEXT:RETURN
1175 REM ***** DESENHA COROA CIRCULAR
1180 A=26:B=35:FORX=10T012:GOSUB1680:A=A-3:B=
B+3:NEXT:FORX=18T043:RESET(X,13):NEXT:A=17:B=
44:FORX=14T016:GOSUB1680:A=A-1:B=B+1:NEXT:FOR
X=15T046:RESET(X,17):NEXT
1190 A=26:B=35:FORX=24T022STEP-1:GOSUB1680:A=
A-3:B=B+3:NEXT:FORX=18T043:RESET(X,21):NEXT:A
=17:B=44:FORX=20T018STEP-1:GOSUB1680:A=A-1:B=
B+1:NEXT
1200 FORX=32T046STEP2:SET(X,17):NEXT:FORX=47T
057STEP2:RESET(X,17):NEXT:FORX=24T017STEP-2:S
ET(30,Y):NEXT:RETURN
1205 REM ***** DESENHA SETOR CIRCULAR
1210 FORX=29T042:SET(X,5):SET(X,29):NEXT:FORX
=25T029:SET(X,6):SET(X,28):NEXT:FORX=42T046:S
ET(X,6):SET(X,28):NEXT:FORX=46T051:SET(X,7):N
EXT:X=19:FORX=0T017:SET(X,Y):X=X+1:NEXT:X=19:
FORX=34T017STEP-1:SET(X,Y):X=X+1:NEXT
1220 FORX=6T017STEP2:SET(37,Y):NEXT
1230 A=21:B=25:FORX=7T011:GOSUB1670:A=A-2:B=B
+1:NEXT:A=21:B=25:FORX=27T023STEP-1:GOSUB1670
:A=A-2:B=B+1:NEXT:FORX=12T031:SET(X,12):SET(X
,22):NEXT:FORX=11T032:SET(X,21):SET(X,13):NEX
T:FORX=11T033:SET(X,14):SET(X,20):NEXT
1240 A=10:B=34:FORX=15T017:GOSUB1670:B=B+1:NE
XT:A=10:B=34:FORX=19T018STEP-1:GOSUB1670:B=B+
1:NEXT:RETURN
1245 REM ***** DESENHA SEGMENTO CIRCULAR
1250 A=21:B=22:FORX=7T011:GOSUB1270:NEXT:A=21
:B=22:FORX=27T023STEP-1:GOSUB1270:NEXT:FORX=1
2T022:SET(X,12):SET(X,22):NEXT:FORX=11T022:SE
T(X,13):SET(X,14):SET(X,20):SET(X,21):NEXT:FO
RX=15T019:FORX=10T022:SET(X,Y):NEXT:NEXT
1260 X=18:FORX=1T017:SET(X,Y):X=X+1:NEXT:X=18
:FORX=33T018STEP-1:SET(X,Y):X=X+1:NEXT:RETURN
1270 GOSUB1670:A=A-2:RETURN
1275 REM ***** DESENHA SETOR DE COROA CIRCULA
R
1280 FORX=29T042:SET(X,5):SET(X,29):NEXT:FORX
=25T029:SET(X,6):SET(X,28):NEXT:FORX=42T046:S
ET(X,28):NEXT:FORX=46T051:SET(X,27):NEXT:FORX
=32T042:SET(X,8):NEXT:FORX=29T032:SET(X,9):NE
XT:FORX=41T045:SET(X,9):NEXT
1290 X=19:FORX=0T017:SET(X,Y):X=X+1:NEXT:X=19
:FORX=34T017STEP-1:SET(X,Y):X=X+1:NEXT:A=21:B
=25:FORX=7T011:GOSUB1670:A=A-2:B=B+1:NEXT:A=2
1:B=25:FORX=27T023STEP-1
1300 GOSUB1670:A=A-2:B=B+1:NEXT:FORX=12T031:S
ET(X,12):SET(X,22):NEXT:FORX=11T032:SET(X,21)
:SET(X,13):NEXT
1310 FORX=11T033:SET(X,14):SET(X,20):NEXT:A=1
0:B=34:FORX=19T018STEP-1:GOSUB1670:B=B+1:NEXT
:A=10:B=34:FORX=15T017:GOSUB1670:B=B+1:NEXT:A
=27:B=29:FORX=11T013:GOSUB1680:A=A-2:B=B+1:NE
XT
1320 A=27:B=29:FORX=23T021STEP-1:GOSUB1680:A=
A-2:B=B+1:NEXT:A=22:B=32:FORX=20T018STEP-1:GO
SUB1680:A=A-1:B=B+1:NEXT
1330 A=22:B=32:FORX=14T016:GOSUB1680:A=A-1:B=
B+1:NEXT:FORX=20T035:RESET(X,17):NEXT:FORX=9T
017STEP2:SET(36,Y):NEXT:X=36:FORX=17T028STEP2
:SET(X,Y):X=X+1:NEXT:RETURN
1335 REM ***** DESENHA ELIPSE
1340 FORX=34T056:SET(X,6):SET(X,28):NEXT:FORX
=26T064:SET(X,7):SET(X,27):NEXT:FORX=20T070:S
ET(X,8):SET(X,26):NEXT:FORX=16T074:SET(X,9):S
ET(X,25):NEXT:FORX=13T077:SET(X,10):SET(X,24)
:NEXT
1350 A=11:B=79:FORX=11T012:GOSUB1670:A=A-2:B=
B+2:NEXT:A=11:B=79:FORX=23T022STEP-1:GOSUB167
0:A=A-2:B=B+2:NEXT:A=8:B=82:FORX=13T016:GOSUB
1670:A=A-1:B=B+1:NEXT:A=8:B=82:FORX=21T018STE
P-1:GOSUB1670:A=A-1:B=B+1:NEXT
1360 FORX=5T090STEP2:SET(X,17):NEXT:FORX=7T02
7STEP2:RESET(45,Y):NEXT:SET(45,30):FORX=20T03
0STEP2:SET(5,Y):NEXT:POKEP+652,88:FORX=47T091
STEP2:SET(X,6):NEXT:POKEP+235,89:RETURN
1365 REM ***** DESENHA PARABOLA
1370 SET(5,20):FORX=17T030STEP2:SET(45,Y):NEX
T:FORX=34T045:SET(X,6):NEXT:FORX=26T045:SET(X
,7):NEXT:FORX=20T045:SET(X,8):NEXT:FORX=16T04
5:SET(X,9):NEXT:A=13:B=45:FORX=10T012:GOSUB16
70:A=A-2:NEXT
1380 A=8:B=45:FORX=13T016:GOSUB1670:A=A-1:NEX
T:FORX=5T045:SET(X,17):NEXT:RETURN
1390 INPUT"Entre com a area ";A:RETURN
1400 INPUT"Entre com a diagonal (D) ";D:RETUR
N
1410 INPUT"Entre com um dos lados ";M:RETURN
1420 INPUT"Entre com o outro lado ";N:RETURN
1430 INPUT"Entre com a altura (h) ";H:RETURN
1440 INPUT"Entre com o lado menor ";L1:RETURN
1450 INPUT"Entre com o lado maior ";L2:RETURN
1460 INPUT"Entre com o valor do raio menor ";
R1:RETURN
1470 INPUT"Entre com o valor do raio maior ";
R2:RETURN
1480 PRINT@704,"":INPUT"Entre com o valor do
angulo alfa ";Z:RETURN
1490 PRINT@704,"":INPUT"Entre com o valor de
X ";X:RETURN
1500 INPUT"Entre com o valor de Y ";Y:RETURN
1510 PRINT@704,STRING$(40,32):RETURN
1520 PRINT@768,STRING$(40,32):RETURN
1530 PRINT@832,STRING$(40,32):RETURN
1540 PRINT@896,STRING$(40,32):RETURN
1550 PRINT@960,STRING$(44,32):RETURN
1560 FORP=16064T016106:POKEP,176:NEXT:FORP=16
192T016234:POKEP,131:NEXT:FORI=1T06:PRINT@768
," 0 resultado e' ";:F
ORD=1T050:NEXT:PRINT@784,USING"###,###,###,##
H,####";V,:FORD=1T0400:NEXT:NEXT:RETURN
1570 PRINT@960,"Tecle << RETURN >> para nov
o calculo ";
1580 A$=INKEY$:IF A$(13) THEN 1580 ELSE RETU
RN
1590 FORD=1T03:SET(X,D,Y):NEXT:X=X-2:RETURN
1600 GOSUB1510:GOSUB1520:GOSUB1530:GOSUB1550:
GOSUB1560:GOSUB1570:GOT040
1610 INPUT"Entre com o valor do lado (a) ";L:
RETURN
1620 INPUT"Entre com o valor do lado (b) ";B:
RETURN
1630 INPUT"Entre com o valor do lado (c) ";C:
RETURN
1640 INPUT"Entre com o valor da altura (h) ";
H:RETURN
1650 INPUT"Entre com o valor do raio (r) ";R:
RETURN
1660 CLS:PRINT@960,"<INFORME VALORES NA MESMA
UNIDADE DE MEDIDA>";:P=15360:RETURN
1670 FORX=ATOB:SET(X,Y):NEXT:RETURN
1680 FORX=ATOB:RESET(X,Y):NEXT:RETURN
1690 PRINT@704,"O VALOR DA AREA DA BASE E'";S
:GOSUB1640:RETURN
1700 PRINT@704,"":GOSUB1650:GOSUB1640:RETURN

```


Muita atenção ao disparar os seus tiros. Você estará enfrentando um inimigo esperto, um micro da família Sinclair. Portanto, comandante, fogo neles!

Batalha Naval

Waldir Ferreira de Arruda



O tradicional jogo *Batalha Naval* ganha novo cenário: a tela dos micros da família Sinclair. As regras, os navios, seus formatos e quantidades são os mesmos (4 submarinos, 4 destroyers, 5 hidroaviões, 2 cruzadores e 1 couraçado) e você será o humano que jogará contra o micro.

Logo depois de rodado, o programa pede o nível de jogo: 1 (fácil), 2 (médio) e 3 (difícil). Em seguida, desenha a tela com os dois campos de batalha, sendo que o campo do lado direito pertence ao micro.

O programa entra em FAST para montar, randomicamente, os navios do micro. Ao final, ele monitora a entrada de dados para montar os navios do jogador (humano). Isto se faz com a digitação de "linhas e colunas" e o programa pede a confirmação para cada coordenada digitada (ao todo, são 38 partes de

navio). Para cada navio a ser montado pelo humano, o micro mostra o formato, o qual, no entanto, não é checado pelo programa. Portanto, podem ser criados novos formatos.

A BATALHA

O humano inicia o jogo — depois de ter desenhado todos os seus navios — tendo direito a três tiros. Ao término destes, o micro diz se foi acertada alguma parte de navio ou se foi apenas água. Para cada parte acertada, há uma inversão de tela simulando uma explosão e, para cada tiro na água, segue-se um bip. Agora, se o tiro cair em alguma parte já acertada, o micro emite três bips e esta parte permanece inversa (não conta pontos).

Logo que o humano termina seus ti-

ros, o micro faz a seguinte pergunta: RECONHECIMENTO (S/N). Se o jogador responde N, o micro lança seus torpedos (se der água, o micro também emite um bip e preenche esta coordenada com um asterisco; se atingir alguma parte, esta ficará piscando, tomando-se inversa). Caso o jogador responda S, passa pela tela a confirmação de quantas partes se compõe cada navio.

A esta altura, inicia-se também uma contagem regressiva de pontos, pois a somatória de todas as partes de cada comandante equivale a 38 partes. O comandante que conseguir chegar até zero ganha a *Batalha Naval*.

DETALHES TÉCNICOS

A listagem em Assembler deve ser digitada com o auxílio do monitor publi-

Bloco Assembler

16514 = 42	16530 = 24
16515 = 12	16531 = 2
16516 = 64	16532 = 198
16517 = 6	16533 = 128
16518 = 22	16534 = 119
16519 = 197	16535 = 16
16520 = 6	16536 = 241
16521 = 32	16537 = 35
16522 = 35	16538 = 193
16523 = 126	16539 = 16
16524 = 254	16540 = 234
16525 = 127	16541 = 201
16526 = 56	16542 = 211
16527 = 4	16543 = 240
16528 = 222	16544 = 201
16529 = 128	

cado em MICRO SISTEMAS número 23. Ela é composta de duas rotinas (é aconselhável, portanto, reservar 31 bytes na primeira linha de REM):

- de 16514 a 16541 – rotina de inversão da tela;
- de 16542 a 16544 – rotina de bip (esta rotina é para o caso do microcomputador CP-200, no qual o programa foi desenvolvido; nos demais equipamentos que não têm bip, esta rotina deve ser omitida, e as linhas que contiverem a instrução RAND USR 16542 não devem ser digitadas).

Como sabemos, uma das técnicas utilizadas para jogar *Batalha Naval* é dar tiros em volta da parte atingida de um navio para tentar afundá-lo todo. No entanto, esta rotina não pôde ser implantada neste programa, pois ela demandaria mais alguns Kbs de memória, coisa de que não se dispunha. Para contornar a situação e tornar a *Batalha* mais competitiva, optei pelos níveis de dificuldade mencionados no início do artigo.

Observação final: para obter o caráter gráfico situado nas linhas 5340, 5520, 5690, 5750 e 5791, digite GRAPHICS SHIFT A. Outra coisa: caso o humano queira desistir, basta digitar dois asteriscos (**) na sua vez de atirar.

Waldir Ferreira de Arruda estudou na FASP (Faculdade de Sistemas de São Paulo). Atualmente trabalha na empresa Incremento Informática S/A., onde ocupa o cargo de Analista de Produção. Possui um CP-200, utilizando-o como hobby.

Batalha Naval

```

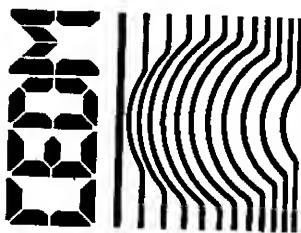
10 REM (31 CARACTERES QUAISQUE
R)
15 RAND
16 REM WALDIR F. ARRUDA
20 DIM M$(15,15)
30 DIM P$(15,15)
36 GOSUB VAL "8500"
40 PRINT AT 7,2;"[O][O][O][O][O][O][O][O][O][O][O][O][O][O][O][O]"
50 PRINT AT 8,2;"[O][O][O][O][O][O][O][O][O][O][O][O][O][O][O][O]"
60 PRINT AT 9,2;"[O][O][O][O][O][O][O][O][O][O][O][O][O][O][O][O]"
70 PRINT AT 10,2;"[O][O][O][O][O][O][O][O][O][O][O][O][O][O][O][O]"
80 PRINT AT 13,6;"[O][O][O][O][O][O][O][O][O][O][O][O][O][O][O][O]"
90 PRINT AT 14,6;"[O][O][O][O][O][O][O][O][O][O][O][O][O][O][O][O]"
100 PRINT AT 15,6;"[O][O][O][O][O][O][O][O][O][O][O][O][O][O][O][O]"
110 PRINT AT 16,6;"[O][O][O][O][O][O][O][O][O][O][O][O][O][O][O][O]"
120 FOR N=1 TO 16
130 LET X=USR 16514
131 FOR A=1 TO 8
132 NEXT A
140 NEXT N
150 CLS
160 PRINT AT 7,5;"HUMANO";AT 7,
21;"MACHO"
200 FOR N=1 TO 15
210 LET Z=165+N
220 PRINT AT 0,N;CHR$ Z;AT 16,N
;CHR$ Z;
230 NEXT N
240 FOR N=17 TO 31
250 LET Z=149+N
260 PRINT AT 0,N;CHR$ Z;AT 16,N
;CHR$ Z;
270 NEXT N
280 FOR N=1 TO 15
290 LET Z=165+N
300 PRINT AT N,0;CHR$ Z;AT N,16
;CHR$ Z;
310 NEXT N
320 PRINT AT 0,0;"[O]";AT 16,0;"[O]";
;AT 0,16;"[O]";AT 16,16;"[O]"
330 FOR X=1 TO 15
340 FOR N=1 TO 15
350 PRINT AT X,N;". ";
360 NEXT N
370 FOR N=17 TO 31
380 PRINT AT X,N;". ";
390 NEXT N
400 NEXT X
405 PRINT AT 18,1;"AGUARDE.. MI
CRO MONTANDO NAVIOS "
406 FOR N=1 TO 50
407 NEXT N
410 FAST
420 FOR H=1 TO 5
430 LET L=INT (RND*13)+1
440 LET C=INT (RND*13)+1
450 LET T=INT (RND*2)+1
460 IF T=1 THEN LET A=3
470 IF T=1 THEN LET B=2
480 IF T=2 THEN LET A=2
490 IF T=2 THEN LET B=3
500 IF C=13 AND L<=12 THEN LET
A=3
510 IF L=13 AND C<=12 THEN LET
B=3
520 IF L=13 THEN LET A=2
530 IF C=13 THEN LET B=2
540 IF L=13 AND C=13 THEN GOTO
0430
541 FOR Z=0 TO A
542 FOR N=0 TO B
543 IF M$(L+Z,C+N)<>" " THEN GO
TO 0430
544 NEXT N
545 NEXT Z
546 FOR Z=0 TO A
547 FOR N=0 TO B
548 LET M$(L+Z,C+N)="*"
549 NEXT N
550 NEXT Z
560 IF L=13 OR C=13 THEN GOTO V
AL "820"
570 IF T=1 THEN GOTO VAL "700"
580 GOSUB VAL "1070"
590 IF T=1 THEN GOTO VAL "650"
600 LET M$(L,C)="[O]"
610 LET M$(L,C+2)="[O]"
620 LET M$(L+1,C+1)="[O]"
630 NEXT H
640 GOTO VAL "1100"
650 LET M$(L,C+1)="[O]"
660 LET M$(L+1,C)="[O]"
670 LET M$(L+1,C+2)="[O]"
680 NEXT H
690 GOTO VAL "1100"
700 GOSUB VAL "1070"
710 IF T=1 THEN GOTO VAL "770"
720 LET M$(L,C)="[O]"
730 LET M$(L+1,C+1)="[O]"
740 LET M$(L+2,C)="[O]"
750 NEXT H
760 GOTO VAL "1100"
770 LET M$(L,C+1)="[O]"
780 LET M$(L+1,C)="[O]"
790 LET M$(L+2,C+1)="[O]"
800 NEXT H
810 GOTO VAL "1100"
820 IF C=13 THEN GOTO VAL "950"
830 GOSUB VAL "1070"
840 IF T=1 THEN GOTO VAL "900"
850 LET M$(L+1,C+2)="[O]"
860 LET M$(L+2,C+1)="[O]"
870 LET M$(L+2,C+3)="[O]"
880 NEXT H
890 GOTO VAL "1100"
900 LET M$(L+1,C+1)="[O]"
910 LET M$(L+1,C+3)="[O]"
920 LET M$(L+2,C+2)="[O]"
930 NEXT H
940 GOTO VAL "1100"
950 GOSUB VAL "1070"
960 IF T=1 THEN GOTO VAL "1020"
970 LET M$(L+1,C+2)="[O]"
980 LET M$(L+2,C+1)="[O]"
990 LET M$(L+3,C+2)="[O]"
1000 NEXT H
1010 GOTO VAL "1100"
1020 LET M$(L+1,C+1)="[O]"
1030 LET M$(L+2,C+2)="[O]"
1040 LET M$(L+3,C+1)="[O]"
1050 NEXT H
1060 GOTO VAL "1100"
1070 LET T=INT (RND*2)+1
1080 RETURN
1100 LET L=INT (RND*14)+1
1110 LET C=INT (RND*14)+1
1120 LET T=INT (RND*2)+1
1130 IF T=1 THEN LET A=1
1140 IF T=1 THEN LET B=5
1150 IF T=2 THEN LET A=5
1160 IF T=2 THEN LET B=1
1170 IF L=14 AND C>10 THEN GOTO
VAL "1100"
1180 IF C=14 AND L>10 THEN GOTO
VAL "1100"
1190 IF L>10 AND T=2 THEN GOTO V
AL "1100"
1200 IF C>10 AND T=1 THEN GOTO V
AL "1100"
1210 FOR Z=0 TO A
1220 FOR N=0 TO B
1230 IF M$(L+Z,C+N)<>" " THEN GO
TO VAL "1100"
1240 NEXT N
1250 NEXT Z
1260 FOR Z=0 TO A
1270 FOR N=0 TO B
1280 LET M$(L+Z,C+N)="*"
1290 NEXT N
1300 NEXT Z
1310 IF L=14 OR C=14 THEN GOTO V
AL "1410"
1320 IF T=1 THEN GOTO VAL "1370"
1330 FOR N=0 TO 4
1340 LET M$(L+N,C)="[O]"
1350 NEXT N
1360 GOTO VAL "1500"
1370 FOR N=0 TO 4

```

```

1380 LET M$(L,C+N)="0"
1390 NEXT N
1400 GOTO VAL "1500"
1410 IF C=14 THEN GOTO VAL "1460"
1420 FOR N=0 TO 4
1430 LET M$(L+1,C+N)="0"
1440 NEXT N
1450 GOTO VAL "1500"
1460 FOR N=1 TO 5
1470 LET M$(L+N,C+1)="0"
1480 NEXT N
1500 FOR R=1 TO 2
1505 LET PI=PEEK 16591
1510 LET L=INT (RND*14)+1
1520 LET C=INT (RND*14)+1
1530 LET T=INT (RND*2)+1
1540 IF T=1 THEN LET A=1
1550 IF T=1 THEN LET B=4
1560 IF T=2 THEN LET A=4
1570 IF T=2 THEN LET B=1
1580 IF L=14 AND C>11 THEN GOTO VAL "1510"
1590 IF C=14 AND L>11 THEN GOTO VAL "1510"
1600 IF L>11 AND T=2 THEN GOTO V AL "1510"
1610 IF C>11 AND T=1 THEN GOTO V AL "1510"
1620 FOR Z=0 TO A
1630 FOR N=0 TO B
1640 IF M$(L+Z,C+N)<>" " THEN GO TO VAL "1510"
1650 NEXT N
1660 NEXT Z
1670 FOR Z=0 TO A
1680 FOR N=0 TO B
1690 LET M$(L+Z,C+N)="*"
1700 NEXT N
1710 NEXT Z
1720 IF L=14 OR C=14 THEN GOTO V AL "1840"
1730 IF T=1 THEN GOTO VAL "1790"
1740 FOR N=0 TO 3
1750 LET M$(L+N,C)="2"
1760 NEXT N
1770 NEXT R
1780 GOTO VAL "1950"
1790 FOR N=0 TO 3
1800 LET M$(L,C+N)="2"
1810 NEXT N
1820 NEXT R
1830 GOTO VAL "1950"
1840 IF C=14 THEN GOTO VAL "1900"
1850 FOR N=0 TO 3
1860 LET M$(L+1,C+N)="2"
1870 NEXT N
1880 NEXT R
1890 GOTO VAL "1950"
1900 FOR N=1 TO 4
1910 LET M$(L+N,C+1)="2"
1920 NEXT N
1930 NEXT R
1950 FOR D=1 TO 3
1960 LET L=INT (RND*14)+1
1970 LET C=INT (RND*14)+1
1980 LET T=INT (RND*2)+1
1990 IF T=1 THEN LET A=1
2000 IF T=1 THEN LET B=2
2010 IF T=2 THEN LET A=2
2020 IF T=2 THEN LET B=1
2030 IF L=14 AND C>13 THEN GOTO VAL "1960"
2040 IF C=14 AND L>13 THEN GOTO VAL "1960"
2050 IF L>13 AND T=2 THEN GOTO V AL "1960"
2060 IF C>13 AND T=1 THEN GOTO V AL "1960"
2070 FOR Z=0 TO A
2080 FOR N=0 TO B
2090 IF M$(L+Z,C+N)<>" " THEN GO TO VAL "1960"
2100 NEXT N
2110 NEXT Z
2120 FOR Z=0 TO A
2130 FOR N=0 TO B
2140 LET M$(L+Z,C+N)="*"
2150 NEXT N
2160 NEXT Z
2170 IF L=14 OR C=14 THEN GOTO V AL "2290"
2180 IF T=1 THEN GOTO VAL "2240"
2190 FOR N=0 TO 1
2200 LET M$(L+N,C)="0"
2210 NEXT N
2220 NEXT D
2230 GOTO VAL "2400"
2240 FOR N=0 TO 1
2250 LET M$(L,C+N)="0"
2260 NEXT N
2270 NEXT D
2280 GOTO VAL "2400"
2290 IF C=14 THEN GOTO VAL "2350"
2300 FOR N=0 TO 1
2310 LET M$(L+1,C+N)="0"
2320 NEXT N
2330 NEXT D
2340 GOTO VAL "2400"
2350 FOR N=1 TO 2
2360 LET M$(L+N,C+1)="0"
2370 NEXT N
2380 NEXT D
2400 FOR S=1 TO 4
2410 LET L=INT (RND*15)+1
2420 LET C=INT (RND*15)+1
2430 IF M$(L,C)=" " THEN GOTO VA L "2450"
2440 GOTO VAL "2410"
2450 IF L=1 OR L=15 OR C=1 OR C=15 THEN GOTO VAL "2510"
2460 IF M$(L-1,C-1)<>"*" AND M$(L-1,C-1)<>" " OR M$(L-1,C)<>"*" AND M$(L-1,C)<>" " OR M$(L-1,C+1)<>"*" AND M$(L-1,C+1)<>" " OR M$(L,C-1)<>"*" AND M$(L,C-1)<>" " OR M$(L,C)<>"*" AND M$(L,C)<>" " OR M$(L,C+1)<>"*" AND M$(L,C+1)<>" " OR M$(L+1,C-1)<>"*" AND M$(L+1,C-1)<>" " OR M$(L+1,C)<>"*" AND M$(L+1,C)<>" " OR M$(L+1,C+1)<>"*" AND M$(L+1,C+1)<>" " THEN GOTO VAL "2410"
2470 GOTO VAL "2760"
2510 IF L=1 AND C=1 THEN GOTO VA L "2600"
2520 IF L=1 AND C=15 THEN GOTO V AL "2620"
2530 IF L=1 AND C>1 AND C<15 THEN GOTO VAL "2640"
2540 IF L=15 AND C>1 AND C<15 THEN GOTO VAL "2660"
2550 IF L=15 AND C=1 THEN GOTO V AL "2680"
2560 IF L=15 AND C=15 THEN GOTO VAL "2700"
2570 IF C=1 AND L>1 AND L<15 THEN GOTO VAL "2720"
2580 IF C=15 AND L>1 AND L<15 THEN GOTO VAL "2740"
2600 IF M$(L,C+1)<>"*" AND M$(L,C+1)<>" " OR M$(L+1,C)<>"*" AND M$(L+1,C)<>" " OR M$(L+1,C+1)<>"*" AND M$(L+1,C+1)<>" " THEN GOT O 2410
2610 GOTO VAL "2760"
2620 IF M$(L,C-1)<>"*" AND M$(L,C-1)<>" " OR M$(L+1,C)<>"*" AND M$(L+1,C)<>" " OR M$(L+1,C-1)<>"*" AND M$(L+1,C-1)<>" " THEN GOT O 2410
2630 GOTO VAL "2760"
2640 IF M$(L,C-1)<>"*" AND M$(L,C-1)<>" " OR M$(L,C+1)<>"*" AND M$(L,C+1)<>" " OR M$(L+1,C-1)<>"*" AND M$(L+1,C-1)<>" " OR M$(L+1,C+1)<>"*" AND M$(L+1,C+1)<>" " OR M$(L,C)<>"*" AND M$(L,C)<>" " OR M$(L+1,C)<>"*" AND M$(L+1,C)<>" " OR M$(L,C+1)<>"*" AND M$(L,C+1)<>" " OR M$(L,C-1)<>"*" AND M$(L,C-1)<>" " OR M$(L-1,C)<>"*" AND M$(L-1,C)<>" " OR M$(L-1,C+1)<>"*" AND M$(L-1,C+1)<>" " THEN GOTO 2410
2650 GOTO VAL "2760"
2660 IF M$(L,C-1)<>"*" AND M$(L,C-1)<>" " OR M$(L,C+1)<>"*" AND M$(L,C+1)<>" " OR M$(L+1,C)<>"*" AND M$(L+1,C)<>" " OR M$(L+1,C-1)<>"*" AND M$(L+1,C-1)<>" " OR M$(L+1,C+1)<>"*" AND M$(L+1,C+1)<>" " OR M$(L,C)<>"*" AND M$(L,C)<>" " OR M$(L+1,C)<>"*" AND M$(L+1,C)<>" " OR M$(L,C+1)<>"*" AND M$(L,C+1)<>" " OR M$(L,C-1)<>"*" AND M$(L,C-1)<>" " OR M$(L-1,C)<>"*" AND M$(L-1,C)<>" " OR M$(L-1,C+1)<>"*" AND M$(L-1,C+1)<>" " THEN GOTO 2410
2670 GOTO VAL "2760"
2680 IF M$(L-1,C)<>"*" AND M$(L-1,C)<>" " OR M$(L-1,C+1)<>"*" AND M$(L-1,C+1)<>" " OR M$(L,C)<>"*" AND M$(L,C)<>" " OR M$(L,C+1)<>"*" AND M$(L,C+1)<>" " OR M$(L,C-1)<>"*" AND M$(L,C-1)<>" " OR M$(L-1,C)<>"*" AND M$(L-1,C)<>" " OR M$(L-1,C+1)<>"*" AND M$(L-1,C+1)<>" " OR M$(L,C)<>"*" AND M$(L,C)<>" " OR M$(L,C+1)<>"*" AND M$(L,C+1)<>" " OR M$(L,C-1)<>"*" AND M$(L,C-1)<>" " OR M$(L-1,C)<>"*" AND M$(L-1,C)<>" " OR M$(L-1,C+1)<>"*" AND M$(L-1,C+1)<>" " THEN GOTO 2410
2690 GOTO VAL "2760"
2700 IF M$(L,C-1)<>"*" AND M$(L,C-1)<>" " OR M$(L,C+1)<>"*" AND M$(L,C+1)<>" " OR M$(L+1,C)<>"*" AND M$(L+1,C)<>" " OR M$(L+1,C-1)<>"*" AND M$(L+1,C-1)<>" " OR M$(L+1,C+1)<>"*" AND M$(L+1,C+1)<>" " OR M$(L,C)<>"*" AND M$(L,C)<>" " OR M$(L+1,C)<>"*" AND M$(L+1,C)<>" " OR M$(L,C+1)<>"*" AND M$(L,C+1)<>" " OR M$(L,C-1)<>"*" AND M$(L,C-1)<>" " OR M$(L-1,C)<>"*" AND M$(L-1,C)<>" " OR M$(L-1,C+1)<>"*" AND M$(L-1,C+1)<>" " THEN GOT O 2410
2710 GOTO VAL "2760"
2720 IF M$(L-1,C)<>"*" AND M$(L-1,C)<>" " OR M$(L-1,C+1)<>"*" AND M$(L-1,C+1)<>" " OR M$(L,C)<>"*" AND M$(L,C)<>" " OR M$(L,C+1)<>"*" AND M$(L,C+1)<>" " OR M$(L,C-1)<>"*" AND M$(L,C-1)<>" " OR M$(L-1,C)<>"*" AND M$(L-1,C)<>" " OR M$(L-1,C+1)<>"*" AND M$(L-1,C+1)<>" " OR M$(L,C)<>"*" AND M$(L,C)<>" " OR M$(L,C+1)<>"*" AND M$(L,C+1)<>" " OR M$(L,C-1)<>"*" AND M$(L,C-1)<>" " OR M$(L-1,C)<>"*" AND M$(L-1,C)<>" " OR M$(L-1,C+1)<>"*" AND M$(L-1,C+1)<>" " THEN GOT O 2410
2730 GOTO VAL "2760"
2740 IF M$(L-1,C)<>"*" AND M$(L-1,C)<>" " OR M$(L-1,C-1)<>"*" AND M$(L-1,C-1)<>" " OR M$(L-1,C+1)<>"*" AND M$(L-1,C+1)<>" " OR M$(L,C)<>"*" AND M$(L,C)<>" " OR M$(L,C+1)<>"*" AND M$(L,C+1)<>" " OR M$(L,C-1)<>"*" AND M$(L,C-1)<>" " OR M$(L-1,C)<>"*" AND M$(L-1,C)<>" " OR M$(L-1,C+1)<>"*" AND M$(L-1,C+1)<>" " OR M$(L,C)<>"*" AND M$(L,C)<>" " OR M$(L,C+1)<>"*" AND M$(L,C+1)<>" " OR M$(L,C-1)<>"*" AND M$(L,C-1)<>" " OR M$(L-1,C)<>"*" AND M$(L-1,C)<>" " OR M$(L-1,C+1)<>"*" AND M$(L-1,C+1)<>" " THEN GOT O 2410
2750 LET M$(L,C)="3"
2760 LET M$(L,C)="3"
2770 NEXT S
2780 SLOW
2790 PRINT AT 18,1;"OK... AGORA E A SUA VEZ"
2800 FOR N=1 TO 40
2810 NEXT N
2815 DIM A$(38,2)
2816 LET A=0
2817 LET HID=6
2820 PRINT AT 18,1;"MONTE 5 HIDROS"
2900 FOR H=1 TO 5
2901 PRINT AT 18,1;"MONTE ";HID-H;" HIDROS"
2910 FOR N=1 TO 3
2920 GOSUB VAL "3150"
2930 NEXT N
2940 NEXT H
2950 PRINT AT 18,1;"MONTE 1 COURACADO: 12345"
2960 FOR N=1 TO 5
2970 GOSUB VAL "3150"
2980 NEXT N
2981 LET CRU=3
2985 PRINT AT 18,1;"MONTE 2 CRUZADORES: 1234"
2990 FOR Z=1 TO 2
2991 PRINT AT 18,1;"MONTE ";CRU-Z;" CRUZADORES: 1234"
3000 FOR N=1 TO 4
3010 GOSUB VAL "3150"
3020 NEXT N
3030 NEXT Z
3035 LET DES=4
3040 PRINT AT 18,1;"MONTE 3 DESTROYERS: 12"
3050 FOR D=1 TO 3
3055 PRINT AT 18,1;"MONTE ";DES-D;" DESTROYERS: 12"
3060 FOR N=1 TO 2
3070 GOSUB VAL "3150"
3080 NEXT N
3090 NEXT D
3095 LET SUB=5
3100 PRINT AT 18,1;"MONTE 4 SUBMARINOS: "
3110 FOR S=1 TO 4
3115 PRINT AT 18,1;"MONTE ";SUB-S;" SUBMARINOS: "
3120 GOSUB VAL "3150"
3130 NEXT S
3140 GOTO VAL "4000"
3150 PRINT AT 19,1;"DIGITE LINHA /COLUMA"
3160 INPUT L$
3170 IF LEN L$<>2 THEN GOTO VAL "3150"
3180 LET L=CODE L$(1)-37
3181 IF L<1 OR L>15 THEN GOTO VA L "3340"
3190 LET C=CODE L$(2)-37
3220 IF C<1 OR C>15 THEN GOTO VA L "3360"

```



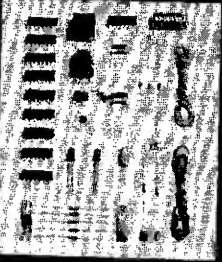
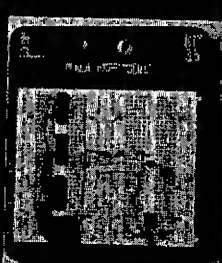
MAIS SUCESSO PARA VOCÊ!

Comece uma nova fase na sua vida profissional.
Os CURSOS CEDM levam até você o mais moderno ensino
técnico programado e desenvolvido no País.

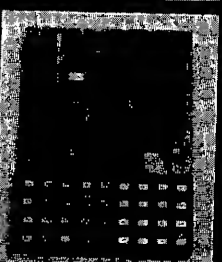
CURSOS DE APERFEIÇOAMENTO

CURSO DE ELETRÔNICA DIGITAL E MICROPROCESSADORES

São mais de 140 apostilas com informações completas e sempre atualizadas. Tudo sobre os mais revolucionários CHIPS. E você recebe, além de uma sólida formação teórica, KITS elaborados para o seu desenvolvimento prático. Garanta agora o seu futuro.

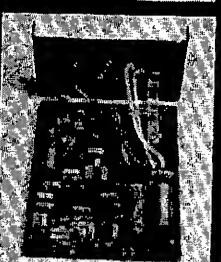
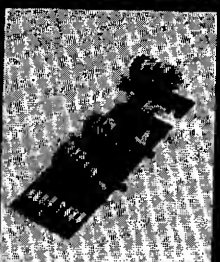
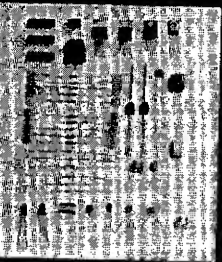


CEDM-20 - KIT de Ferramentas.
CEDM-78 - KIT Fonte de Alimentação 5v/1A. CEDM-35 KIT Placa Experimental
CEDM-74 - KIT de Componentes.
CEDM-80 MICROCOMPUTADOR Z80 ASSEMBLER.



CURSO DE ELETRÔNICA E AUDIO

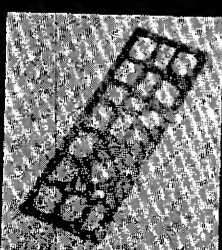
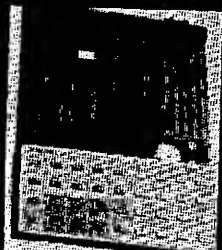
Métodos novos e inéditos de ensino garantem um aprendizado prático muito melhor. Em cada nova lição, apostilas ilustradas ensinam tudo sobre Amplificadores, Caixas Acústicas, Equalizadores, Toca-discos, Sintonizadores AM/FM, Gravadores e Toca-Fitas, Cápsulas e Fonocaptadores, Microfones, Sonorização, Instrumentação de Medidas em Audio, Técnicas de Gravação e também de Reparação em Audio.



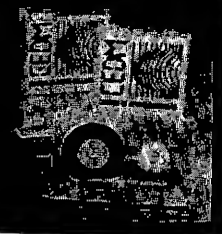
CEDM-1 - KIT de Ferramentas. CEDM-2 - KIT Fonte de Alimentação + 15-15/1A. CEDM-3 - KIT Placa Experimental
CEDM-4 - KIT de Componentes. CEDM-5 - KIT Pré-amplificador Estéreo. CEDM-6 - KIT Amplificador Estéreo 40w.

CURSO DE PROGRAMAÇÃO EM BASIC

Este CURSO, especialmente programado, oferece os fundamentos de Linguagem de Programação que domina o universo dos microcomputadores. Dinâmico e abrangente, ensina desde o BASIC básico até o BASIC mais avançado, incluindo noções básicas sobre Manipulação de Arquivos, Técnicas de Programação, Sistemas de Processamento de Dados, Teleprocessamento, Multiprogramação e Técnicas em Linguagem de Máquina, que proporcionam um grande conhecimento em toda a área de Processamento de Dados.



KIT CEDM Z80
BASIC Científico
KIT CEDM Z80
BASIC Simples.
Gabarito de Fluxograma
E-4. KIT CEDM SOFTWARE
Fitas Cassete com Programas.



GRÁTIS

Você também pode ganhar um MICROCOMPUTADOR.

Telefone (0432) 23-9674 ou coloque hoje mesmo no Correio o cupom CEDM.

Em poucos dias você recebe nossos catálogos de apresentação.

CEDM

Avenida São Paulo, 718 - Fone (0432) 23-9674.
CAIXA POSTAL 1642 - CEP 86100 - Londrina - PR
CURSO DE APERFEIÇOAMENTO POR CORRESPONDÊNCIA

Solicito o mais rápido possível informações sem compromisso sobre o CURSO de

Nome.

Rua.

Cidade.

Bairro. CEP.


```

3230 IF P$(L,C) <> " " THEN GOTO V
AL "3380"
3240 PRINT AT L,C; " "
3250 PRINT AT 19,1; "(S/N) - CONF
IRME - POR FAVOR "
3270 IF INKEY$="S" THEN GOTO VAL
"3400"
3280 IF INKEY$="N" THEN GOTO VAL
"3300"
3290 GOTO VAL "3270"
3300 PRINT AT L,C; " "
3320 PRINT AT 19,1; "
"
3330 GOTO VAL "3150"
3340 PRINT AT 19,1; "LINHA INVALI
DA "
3341 FOR T=1 TO 10
3342 NEXT T
3350 GOTO VAL "3150"
3360 PRINT AT 19,1; "COLUNA INVAL
IDA "
3361 FOR T=1 TO 10
3362 NEXT T
3370 GOTO VAL "3150"
3380 PRINT AT 19,1; "CAMPO JA OCU
PADO "
3381 FOR T=1 TO 10
3382 NEXT T
3390 GOTO VAL "3150"
3400 PRINT AT 19,1; "
"
3410 LET P$(L,C)=" "
3411 LET A=A+1
3412 LET A$(A)=L$
3420 RETURN
4000 PRINT AT 18,1; "MUITO BEM...
VOCE TEM 3 TIROS."
4010 PRINT AT 20,1; "E LOGO EM SE
GUIDA DAREI OS MEUS"
4011 FOR N=1 TO 30
4012 NEXT N
4020 FOR N=1 TO 3
4030 PRINT AT 17+N,1; "
"
4040 NEXT N
5000 LET M=0
5010 LET P=M
5020 DIM I(3)
5030 DIM O(3)
5035 DIM W$(3,2)
5040 FOR N=1 TO 3
5050 PRINT AT 18,1; "INFORME LINH
A E COLUNA >JUNTOS<"
5060 INPUT N$
5065 IF N$="***" THEN GOTO 8525
5070 IF LEN N$ < 2 THEN GOTO VAL
"5160"
5080 LET L=CODE N$(1 TO 1)-37
5090 IF L<1 OR L>15 THEN GOTO VA
L "5160"
5100 LET C=CODE N$(2 TO 2)-37
5110 IF C<1 OR C>15 THEN GOTO VA
L "5160"
5120 PRINT AT L,C+16; " "
5130 LET I(N)=L
5140 LET O(N)=C
5145 LET W$(N)=N$
5150 GOTO VAL "5200"
5160 PRINT AT 18,1; ">>>>> TIRO I
NVALIDO <<<<<< "
5170 FOR Y=1 TO 10
5180 NEXT Y
5190 GOTO VAL "5050"
5200 NEXT N
5205 FOR N=1 TO 3
5210 IF M$(I(N),O(N)) <> " " AND M
$(I(N),O(N)) <> " " THEN GOTO 5260
5220 LET M$(I(N),O(N))=" "
5230 PRINT AT 17+N,1; "TORPEDO ";
N; " AGUA "
5240 NEXT N
5241 FOR Z=1 TO 30
5242 NEXT Z
5250 GOTO VAL "5590"
5260 LET X=CODE M$(I(N),O(N))
5270 IF X=8 THEN GOTO VAL "5330"
5280 IF X=173 THEN GOTO VAL "542
0"
5290 IF X=168 THEN GOTO VAL "544
0"

```

```

5300 IF X=191 THEN GOTO VAL "546
0"
5310 IF X=184 THEN GOTO VAL "548
0"
5320 IF X=169 THEN GOTO VAL "550
0"
5330 PRINT AT 17+N,1; "TORPEDO ";
N; " PARTE JA ATINGIDA "
5340 PRINT I(N),O(N)+16; " "
5350 FOR B=1 TO 3
5360 RAND USR 16542
5370 FOR Z=1 TO 5
5380 NEXT Z
5390 NEXT B
5400 NEXT N
5410 GOTO VAL "5590"
5420 PRINT AT 17+N,1; "TORPEDO ";
W$(N); " >> HIDRO << "
5430 GOTO VAL "5510"
5440 PRINT AT 17+N,1; "TORPEDO ";
W$(N); " > COURACADO < "
5450 GOTO VAL "5510"
5460 PRINT AT 17+N,1; "TORPEDO ";
W$(N); " >-> CRUZADOR < "
5470 GOTO VAL "5510"
5480 PRINT AT 17+N,1; "TORPEDO ";
W$(N); " > SUBMARINO < "
5490 GOTO VAL "5510"
5500 PRINT AT 17+N,1; "TORPEDO ";
W$(N); " > DESTROYER < "
5510 PRINT AT I(N),O(N)+16; " "
5520 LET M$(I(N),O(N))=" "
5530 FOR W=1 TO 6
5540 RAND USR 16514
5550 NEXT W
5560 LET P=P+1
5561 PRINT AT 21,1; "HUMANO =";38
-P,AT 21,15; "MICRO =";38-M; " "
5570 IF P=38 THEN GOTO VAL "5900
"
5580 NEXT N
5581 FOR N=1 TO 50
5582 NEXT N
5590 FOR N=1 TO 3
5600 PRINT AT 17+N,1; "
"
5610 NEXT N
5611 PRINT AT 18,1; "RECONHECIMEN
TO? (S/N) "
5612 IF INKEY$="S" THEN GOTO VAL
"8000"
5613 IF INKEY$="N" THEN GOTO VAL
"5620"
5614 GOTO VAL "5612"
5620 PRINT AT 18,1; "E MINHA VEZ
... LA VAI CHUMBO"
5630 FOR N=1 TO 10
5640 NEXT N
5650 PRINT AT 18,1; "
"
5655 FAST
5660 FOR N=1 TO 3
5661 IF NI=1 THEN GOTO VAL "5670
"
5662 IF NI=2 THEN GOTO VAL "5664
"
5663 IF NI=3 THEN GOTO VAL "5669
"
5664 IF M<P THEN GOTO VAL "5741"
5665 GOTO VAL "5670"
5669 IF M<=P THEN GOTO VAL "5741
"
5670 LET L=INT (RND*15)+1
5680 LET C=INT (RND*15)+1
5690 IF P$(L,C)=" " OR P$(L,C)="
" THEN GOTO VAL "5670"
5700 IF P$(L,C)=" " THEN GOTO VA
L "7000"
5710 LET P$(L,C)=" "
5720 PRINT AT L,C; " "
5725 RAND USR 16542
5730 NEXT N
5735 SLOW
5740 GOTO VAL "5830"
5741 LET SA=INT (RND*38)+1
5742 IF A$(SA)="***" THEN GOTO VA
L "5741"
5743 LET L=CODE A$(SA)-37
5744 LET C=CODE A$(SA,2 TO )-37
5745 LET A$(SA)="***"

```

```

5750 LET P$(L,C)=" "
5755 SLOW
5760 PRINT AT L,C; " "
5770 FOR W=1 TO 10
5771 PRINT AT L,C; " "
5780 PRINT AT L,C; " "
5781 RAND USR 16542
5790 NEXT W
5791 PRINT AT L,C; " "
5800 LET M=M+1
5801 PRINT AT 21,1; "HUMANO =";38
-P,AT 21,15; "MICRO =";38-M; " "
5810 IF M>=38 THEN GOTO VAL "600
0"
5815 FAST
5820 NEXT N
5825 SLOW
5830 PRINT AT 18,1; " >>> HUMANO
JOGA <<<< "
5840 FOR N=1 TO 10
5850 NEXT N
5860 PRINT AT 18,1; "
"
5870 GOTO VAL "5020"
5900 PRINT AT 18,1; " >>> PERDI
A BATALHA <<< "
5910 PRINT AT 19,1; "MEUS NAVIOS
FORAM DESTRUIDOS"
5920 PRINT AT 20,1; " R E N D
O - M E "
5930 FOR N=1 TO 50
5940 NEXT N
5960 CLS
5970 GOTO VAL "8530"
6000 PRINT AT 18,1; "VOCE FOI RED
UZIDO A CINZAS"
6010 PRINT AT 19,1; "NAO CONSEGUI
U RESISTIR AOS MEUS"
6020 PRINT AT 20,1; "TIROS CERTEI
ROS ... HA HA HA "
6030 GOTO VAL "5930"
7000 FOR W=1 TO 38
7010 IF CHR$( (L+37)+CHR$( (C)+
37))=A$(W) THEN LET A$(W)="***"
7020 NEXT W
7030 GOTO 5750
8000 PRINT AT 19,1; "COURACADO 5
PARTES"
8010 GOSUB VAL "8100"
8020 PRINT AT 19,1; "CRUZADOR 4
PARTES"
8030 GOSUB VAL "8100"
8040 PRINT AT 19,1; "DESTROYERS 2
PARTES"
8050 GOSUB VAL "8100"
8060 PRINT AT 19,1; "SUBMARINO 1
PARTE "
8070 GOSUB VAL "8100"
8080 GOTO VAL "5590"
8100 FOR W=1 TO 15
8110 NEXT W
8120 RETURN
8500 PRINT "....." "ESCOLHA O NI
VEL DE JOGO", "1 - FACIL", "2 -
MEDIO", "3 - DIFICIL"
8501 INPUT NI
8510 IF NI<1 OR NI>3 THEN GOTO V
AL "8501"
8515 CLS
8520 RETURN
8525 CLS
8530 LET Z$="ABCDEFGHIJKLMNO"
8540 PRINT AT 1,5; Z$
9000 FOR W=1 TO 15
9005 LET X=165+W
9010 PRINT AT W+1,4; CHR$(X); M$(W)
;CHR$(X)
9020 NEXT W
9025 PRINT AT 17,5; Z$
9030 PRINT AT 19,5; "NAVIOS DO MI
CRO"
9040 PRINT AT 21,1; "P/ OUTRA BAT
ALHA APORTE RUN"
9045 STOP
9050 SAVE "BATALHA NAVAL"
9060 RUN

```

Mensagem de erro

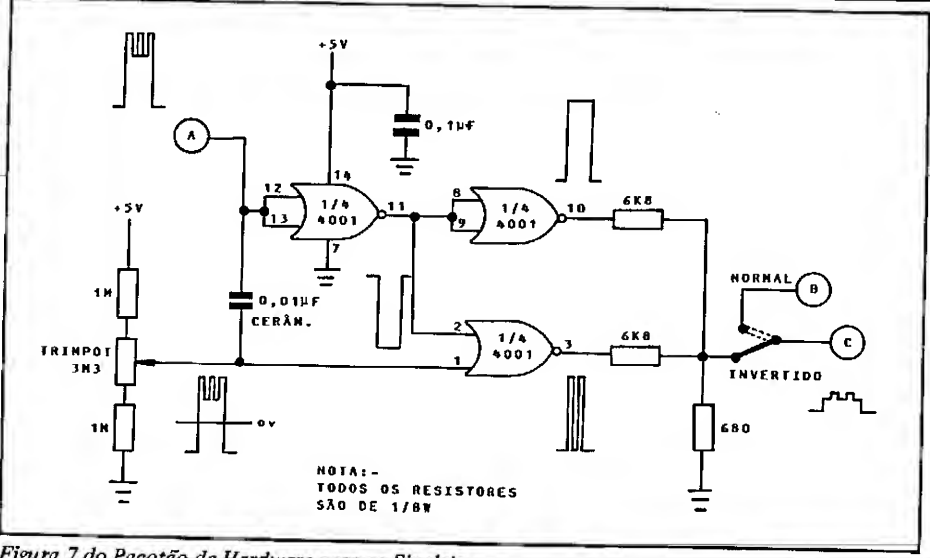
MS Nº	NA PÁGINA	CORREÇÃO
27	75, no programa <i>Projeto de Antenas Direcionais</i> , troque as linhas 170 e 590 que foram publicadas por:	<pre> 170 LET REF=INT(500/F*30+0.5)/100 590 FOR I=INI TO FIM </pre>
29	15, no programa <i>Vigas Contínuas</i> , troque as linhas 480, 510, 550, 635, 780, 940, 1160 e 1200 que foram publicadas por:	<pre> 480 R=-(U(1,2)*X+U(1,3))/U(1,1):R(1)=R:R(2)=X:MF=U(2,1)*R+U(2,2)*X+U(2,3):GOTO 520 510 R=(Y-U(2,2)*X-U(2,3))/U(2,1):R(1)=R:R(2)=X:MF=Y 550 PRINT"MOM. NEGATIVO = ",INT(R(2)) 635 PRINT:PRINT:PRINT"MOM. FINAL = ",,INT(MF) 780 PRINT"QE = ",,INT(Q) 940 IF D<0 THEN GOTO 1010 1160 G=P(A) : Q=2*Z(1)*A(G)+Z(2) : PRINT "QD= ",,INT(Q) 1200 PRINT"M+ = ",INT(X(4)) </pre>
29	42, no programa <i>Linha de Influência</i> , troque as linhas 75, 430 e 460 que foram publicadas por:	<pre> 75:INPUT "SIMP(0), ENG(0.5) = ";F 430:Y=H*(M*R-S)/(1-H*M) 460:Y=M*(H*S-R)/(1-M*H) </pre>
29	65, no programa <i>Catálogo</i> , troque a linha 38400 que foi publicada por:	<pre> 38400 IFJ>JNMP%(C1)THENC3(7)=J-FNMP%(C1):J=FNMP%(C1):GOSUB 38500:C3(1)=C3(1)-J:C3(6)=C3(6)+J:J=C3(7):GOTO 38400 </pre>
29	77, no programa <i>Marcus</i> , troque a linha 200 que foi publicada por:	200 .5
30	38, na matéria <i>Pacotão de hardware para os Sinclair</i> , a figura 7 foi publicada faltando alguns dados; veja a correta:	 <p style="text-align: center;">NOTA: - TODOS OS RESISTORES SÃO DE 1/8W</p>
31	15, do programa <i>Editor de Textos</i> , as últimas linhas da listagem foram suprimidas; acrescente, após a linha 2360, as seguintes linhas:	<pre> 2370 IF J=TM THEN 2390 2380 IF C\$>=" " AND C\$<="z" THEN R\$=R\$+C\$:J=J+1:GOTO 2410 2390 IF C\$=CHR\$(13) THEN PRINT@K+J," ";:RETURN 2400 IF C\$=CHR\$(8) THEN IF J>0 THEN J=J-1:R\$=LEFT\$(R\$,J) ELSE 2360 ELSE 2360 2410 PRINT@K,R\$;CHR\$(CR);" "; 2420 GOTO 2360 </pre>

Figura 7 do Pacotão de Hardware para os Sinclair

Mantenha seus bytes sob controle

Carlos Alberto Diz

Todos conhecemos o byte e o tratamos com a familiaridade de um velho amigo. Sendo assim, sabemos que um byte é um conjunto de bits (oito) e que um bit é algo que só pode ter dois valores ou estados: 1 (ligado ou setado) ou 0 (desligado ou ressetado). Um bit é, portanto, o menor elemento de informação num computador.

O valor de um byte é determinado pelo estado dos bits que o compõem e pela ordem em que se encontram, pois os bits não são todos iguais. Cada bit, por sua posição dentro do byte, corresponde a uma potência de 2, na seguinte ordem (note que os bits estão numerados da direita para a esquerda, começando com o número 0 e terminando com o número 7):

Bit nº:	7	6	5	4	3	2	1	0	
x	:	7	6	5	4	3	2	1	0
2 x :	128	64	32	16	8	4	2	1	

O valor de um byte é a soma dos valores (potências de 2) individuais dos bits que nele encontram-se setados (lembramos que um número elevado à potência de 0 é igual a 1). Por exemplo:

Bit nº :	7	6	5	4	3	2	1	0
2 x :	128	64	32	16	8	4	2	1

Byte A:	0	1	1	0	0	1	1	1	= 103
Byte B:	1	1	0	0	0	0	0	0	= 192
Byte C:	1	1	1	1	1	1	1	1	= 255

A conclusão é que o valor de um byte é obtido através da soma dos valores correspondentes aos bits que estejam setados. Conforme a tabela que acabamos de ver, o byte A é, portanto, igual a $64+32+4+2+1$, o byte B é igual a $128+64$ e assim por diante. (É interessante notar que o valor do byte C, que tem todos os seus bits setados, é igual a 255, ou seja, $2^8-1 = 256-1 = 255$. Isto significa dizer que podemos calcular o valor máximo representável por um número x de bits como sendo 2^x-1).

Segundo as regras de combinações e permutações, oito coisas que possam ter dois estados cada uma permitem 256 combinações diferentes, como é o nosso caso. Aí alguns vão dizer: "Errou! Errou! Se o valor máximo de um byte é 255, como pode ter 256 combinações?" E eu direi: "Não se esqueçam do zero! Do 0 ao 255 são 256 passos, certo?"

SETANDO E RESSETANDO

Uma das tantas conclusões que se pode tirar da observação de números binários é que cada bit representa um valor que é uma função de sua posição dentro do byte. Por exemplo, o terceiro bit (contando sempre da direita para a esquerda) representa o valor 4 e o sétimo o valor 64.

Suponhamos ter um byte com valor 90 (01011010). Se conseguirmos ressetar o bit número 3, transformaremos o byte 90 em 82 (01010010), pois a única diferença entre 90 e 82 é o bit número 3 que, não por acaso, representa o valor 8, ou seja, $90-82 = 8$. Ressetar um bit significa portanto subtrair do byte o valor correspondente a este bit, enquanto setar o bit equivale a somar.

Até aqui nenhuma novidade, mas como fazer para setar ou ressetar o bit? Vamos, então, ao que afinal nos interessa: como manipular estes bits, ou seja, como setar ou ressetar um bit dentro de um byte. Para isto, será necessário refrescar um pouco a memória sobre as propriedades das funções lógicas AND, OR e NOT.

A figura 1 mostra as tabelas de resultados destas operações, que, apesar de aplicáveis no micro a bytes inteiros, agem, na realidade, a nível de bit. Por exemplo: a função AND, quando aplicada entre dois bytes (A AND B), produzirá um terceiro (C) cujos bits serão o resultado de um AND entre os bits de A e os de B, um a um, segundo as regras que constam da figura 1.

Usando os mesmos bytes que usamos anteriormente, se $A=90$ e $B=82$, então $A \text{ AND } B = C$ e $C=82$ (figura 2). Maravilha! Conseguimos encontrar uma maneira de ressetar o bit número 3 em A, e apesar de não ser exatamente a solução, aqueles que curtem um *suspense* já têm aqui uma boa dica.

Se eu quiser ressetar o bit número 1 de 18 (00010010), devo então *ANDeá-lo* (que o Aurélio me perdoe!) com 16, pois $18 \text{ AND } 16 = 16$ (isto é, 18 com o bit 1 ressetado). "É, mas se eu souber de antemão qual o valor de B para transformar A em B, por que não uso B de uma vez e pronto?", dirão aqueles que sempre dizem estas coisas e com razão, pois o que precisamos é de uma técnica que seja independente do byte original.

X	Y	X AND Y	X	Y	X OR Y	X	NOT(X)
1	1	1	1	1	1	1	0
0	1	0	0	1	1	0	1
1	0	0	1	0	1		
0	0	0	0	0	0		

Figura 1

A = 01011010=90
 B = 01010010=82
 A AND B = C = 01010010=82

Figura 2

E realmente a dica é boa, já que um momento de reflexão revela que conseguimos ressetar o bit que queríamos porque fizemos um AND dele com 0, e o resto do byte ficou inalterado porque todos os demais bits foram ANDados com 1 (confira você mesmo). A conclusão é que, para ressetar um bit num byte determinado, é preciso ANDear este último com outro cujos bits estejam todos setados, exceto o que corresponde à posição do bit que queremos ressetar. Em outras palavras (ou números): para ressetar o bit número 3 ($2^3 = 8$) de 90 precisamos ANDe-lo com 247 ($255 - 8 = 247$), pois $01011010 \text{ AND } 11110111 = 01010010$, certo? E, da mesma forma, para setar este bit de novo, precisamos ORe-lo com 8, pois $01010010 \text{ OR } 00001000 = 01011010$. E, finalmente, como $\text{NOT}(00001000) = 11110111$, então $90 \text{ AND}(255 - 8) = 90 \text{ AND } \text{NOT}(8)$.

Fazendo algumas experiências, descobrimos, adicionalmente, que se chamarmos de X um byte que consista de um só bit na posição correspondente ao bit em A que desejamos manipular, então:

X se o bit em questão estiver setado
 A AND X = 0 se o bit em questão estiver ressetado

Em resumo:

- 1 - a função AND permite verificar o estado de um bit;
- 2 - a função AND NOT permite ressetar um bit;
- 3 - a função OR permite setar um bit.

E aqui os sequazes de São Tomé se questionarão: "Será este cara mais um daqueles loucos que gostam de descobrir maneiras de contrariar o computador e fazer o impossível só pelo prazer de dizer que está feito, ou este papo de mexer nos bits serve para alguma coisa de útil?" E a estes eu respondo: as grandes invenções não surgem em função de uma necessidade, mas sim vice-versa, porém desta vez já existe uma aplicação prática para tudo que acabo de revelar...

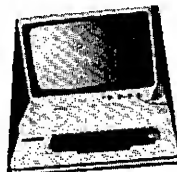
EXEMPLOS DE APLICAÇÃO

Como primeiro exemplo, suponhamos que em um determinado programa espere-se do operador, pela razão que seja, o input de uma letra, por exemplo, A maiúsculo. Se o teclado do computador tiver maiúsculas e minúsculas e o programa não especificar que espera uma letra maiúscula, o operador estará sujeito a teclear um a ou um A, dependendo mais da inspiração do momento do que de uma consideração lógica sobre as consequências de uma decisão ligeira. E, segundo a lei de Murphy, ele sempre teclará a letra errada e o seu programa poderá então fazer coisas incríveis...

Contando com esta inevitável insensibilidade do operador, você poderá fazer com que o programa aceite indiferentemente minúsculas e maiúsculas. Para tanto, você poderá usar uma das 213 diferentes soluções encontradas para este problema pelos analistas do American Institute of Informatic Idiosyncrasies (AIII), como por exemplo:



INFORMÁTICA AO ALCANCE DE TODOS



ENTRE NA ERA
DOS COMPUTADORES

TAXA
UNICA

Cr\$ 30.000,00
SEM MENSALIDADE

- Introdução ao Processamento de Dados • Digitação • Basic
- Cobol • RPG • PL 1 • Fortran
- Assembler • Mumps

227-7417 - 521-4936 - 267-9261

Copa — Centro — Tijuca — Meier
Madureira — N. Iguazu — Caxias

DATABERA^R — PROCESSAMENTO
DE DADOS Rio de Janeiro

O sucesso no Micro-Festival 84,
da qualidade maior,
do preço menor e
das vendas realizadas,

agora tem um novo nome:



e um novo endereço:

SACCO
computer store

A Sacco Computer Store, está agora em novas e amplas instalações com sua variada linha de computadores e periféricos, destacando-se o novo micro-computador CRAFT II plus*, totalmente compatível com D.O.S. 3.3 e CP/M.

Conheça todas as características e vantagens do CRAFT II plus, também compatível com IVANITA**, em nosso novo endereço:

Al. Gabriel Monteiro da Silva, 1229 - J. Paulistano
São Paulo - Tels.: (011) 853-5520 e 280-4778

(*) CRAFT II plus é marca registrada da MICROCRAFT Ind. e Com. Ltda.
(**) IVANITA - Gerador de Caracteres para língua portuguesa


```

10 RP$=STRING$(20,CHR$(0)):REM reseta todos os bits dos 20 bytes que compoem RP$
20 FOR I=0 TO 159: REM 160 vezes, uma para cada resposta
25 BY=INT(I/8)+1: BI=((I/8)-INT(I/8))*8
30 PRINT"Indique a resposta (S/N) ":
40 X$=INKEY$:IF X$="" THEN 40 ELSE X$=CHR$(ASC(X$) AND NOT(32))
50 ON INSTR("SN",X$) GOTO 70,80:REM Aceita somente S ou N
60 GOTO 40:REM Volta a pedir input caso nao tenha sido S nem N
70 MID$(RP$,BY,1)=CHR$(ASC(MID$(RP$,BY,1) OR (2^BI)): REM Seta o bit se for Sim
80 NEXT I:REM Vem direto para aqui se for NAO (os bits foram ressetados no inicio)

```

Figura 3

- dizer ao operador que tipo de letra você quer (pouco imaginativo);
- duplicar o código de resposta (trabalhoso e ineficiente);
- fazer um IF para ver se a letra é minúscula e transformá-la em maiúscula se realmente for (muito pouco elegante e lento).

Você pode escolher qualquer uma, ou então optar pela solução mais rápida e eficiente: fazer um AND da letra *inputada* com NOT(32) para transformá-la em maiúscula se for minúscula, ou deixá-la como está, se for maiúscula.

Gostou? Bom, deixe-me explicar melhor. Supondo que a letra *inputada* está em X\$, então os comandos são: X\$=CHR\$(ASC(X\$) AND NOT(32)), e X\$ será sempre uma letra maiúscula. O segredo está em lembrar que a diferença entre os códigos ASCII de uma letra minúscula e maiúscula é sempre 32 (ASC(a)= 97 e ASC(A)= 65), e em perceber que 32 é o valor do bit 5 e que, portanto, é suficiente ressetar este bit para subtrair 32 de uma letra minúscula e transformá-la em maiúscula.

Já se a letra for maiúscula, o bit 5 já estará ressetado e nossa operação não fará nenhuma diferença.

Suponhamos agora que você queira guardar em disco as respostas de um estudo sobre "Hábitos dos programadores de microcomputadores do Brasil", feito por meio de formulários, sendo que cada formulário contém 160 respostas do tipo "sim ou não".

Se você guardar cada resposta como uma letra (S ou N), estará usando um byte para cada resposta, e assim cada formulário ocupará 160 bytes de memória. Porém, pensando bem, se uma resposta só pode ser SIM ou NÃO, então ela só tem dois estados, exatamente como um bit.

Um bit setado poderia equivaler a um SIM e um bit ressetado a um NÃO. Desta forma, um só byte poderia conter oito respostas, e o formulário ocuparia só 20 bytes em vez de 160, o que representa uma economia não desprezível de 87.5%.

Mas como fazer isso? Bom, se cada byte contém oito respostas, a resposta número X (começando a contar de zero) caberá no byte número BY=INT(X/8)+1 e o bit correspondente será o BI=((X/8)-INT(X/8))*8. Faça um OR de BY com (2^BI) se a resposta X for SIM, ou faça um AND de BY com NOT(2^BI) se a resposta for NÃO.

A figura 3 mostra um simples programinha que faz exatamente isso num TRS-80, inclusive utilizando o macete do primeiro exemplo na linha 140. Usando este programa, as 160 respostas estarão todas contidas na string RP\$, a qual você poderá guardar em disco, fita ou sei lá... Para decodificar RP\$ e obter de volta as respostas, use a função AND para testar os bits um de cada vez, conforme expliquei.

OBSERVAÇÕES FINAIS

Este programa só funciona com BASIC de disco por causa das funções INSTR (linha 50) e MID\$ (linha 70), mas se você não dispõe destas funções, substitua as linhas 50 e 60 como segue:

```

50 IF X$ = "S" THEN 70 ELSE IF X$ = "N" THEN 80
60 Y$ = RIGHT$(LEFT$(RP$,BY),1):RP$ = LEFT$(RP$,BY-1)+CHR$(ASC(Y$) OR (2^BI))+RIGHT$(RP$,LEN(RP$)-BY)

```

É importante também lembrar que as funções lógicas AND, OR e NOT só funcionam a nível de bits no BASIC da Microsoft (que eu saiba) e com certeza não agem da mesma forma no Applesoft, o que impede que os *Apple-amigos* usem estas dicas. De qualquer forma, podem ficar sabendo que, se instalarem CP/M e M-BASIC, poderão manipular seus bits.



Carlos Alberto Diz é formado em Engenharia Eletrônica pela Universidade de Dundee, Escócia, e possui Mestrado em Administração de Empresas pelo INSEAD - Instituto Europeu de Administração de Empresas, Fontainebleau, França. Atualmente é sócio-gerente da Compusystems do Rio de Janeiro, uma system-house dedicada à consultoria informática e confecção de software sob medida.

SoftKristian®

Revendedores Autorizados

Rio de Janeiro

Seletrônica
República do Libano, 25-A
Rio de Janeiro - RJ
CEP: 20061

Gachet
R. Dr. Bujack, 25 S/5
Nova Friburgo - RJ
tel.: 22.4208

VGC
Av. Brasil, 10 S/07
Araucária - RJ
CEP: 28970

ENTRELVROS
Av. Rio Branco, 156 - térreo
Rio de Janeiro - RJ

MCS
Visa de Pirajó, 303/217
Rio de Janeiro - RJ
tel.: 267.8597

Pernambuco

Eletrônica Babel
R. Porto Alegre, 112
Caruaru - PE
CEP: 55100

Alagoas

Expoente
Av. Siqueira Campos, 838
Maceió - AL
tel.: (082) 223.3979

São Paulo

Imarés
Av. dos Imarés, 457
São Paulo - SP
tel.: 61.4049 - 61.0946

Fatolao
R. Boa Vista, 314 - 3º andar
São Paulo - SP
tel.: 35.7131 R/32

Memocards
R. Amador Bueno, 855
Ribeirão Preto - SP
tel.: (016) 636.0586

Folopetia

Alameda Jurúá, 434
São Paulo - SP
tel.: 421.5211

Ritz

R. Frei Caneca, 7
Santos - SP
tel.: 35.1792

Computerland

Av. Angélica, 1996
São Paulo - SP
CEP: 01228

Livraria Poliedro

R. Augusta, 704
São Paulo - SP
tel.: 221.6764

RC Microcomputadores
Av. Estados Unidos, 983
Piracicaba - SP
tel.: 33.7018

Rio Grande do Sul

Advancing
R. Andradas, 1560 galeria
Malcon 518 Porto Alegre - RS
tel.: 26.8246

J.H. Santos

R. Otávio Rocha, 41
Porto Alegre - RS
CEP: 90000

India Center

R. Floriano Peixoto, 1112 conj.
3343 Santa Maria - RS
tel.: (055) 221.7120

Gereima Ltda.

Av. Júlio de Castilhos, 1872
Caxias do Sul - RS
tel.: 221.4299

Nordemag

Av. Júlio de Castilhos, 3240
Caxias do Sul - RS
tel.: 221.3516

Microomega

R. Júlio de Castilhos, 441 -
1º andar Novo Hamburgo - RS
tel.: (0512) 93.4721

Bahia

Oficina
Shopping Center Itaipara
140 - 1º piso
Salvador - BA
tel.: (071) 248.6666

Santa Catarina

Supermicro Show
R. dos Ilheus, 40 II 6
Florianópolis - SC
tel.: 22.8770

Paraná

Computique
Av. Batel, 1750
Curitiba - PR
tel.: 243.4731

Madison

Av. Mal. Deodoro, 311
Curitiba - PR
tel.: 224.3422

Minas Gerais

Computonix
R. Sergipe, 1422
Belo Horizonte - MG
tel.: (031) 225.3305

Eletródio

R. Aquiles Loba, 441-A
Belo Horizonte - MG
tel.: (031) 222.8903

Micro Pócos

R. Assis Figueiredo, 1072
Pócos de Caldas - MG
tel.: (035) 721.1883

Blow-Up

Av. Floriano Peixoto, 396
Uberlândia - MG
tel.: 235.1413 - 235.7359

Brasília

Digitec
SCLN 302 BIA II.63
Brasília - DF
tel.: (061) 225.4534

* CREDENCIAMOS NOVOS REVENDEDORES PARA TODO O BRASIL

Os Kits de Micro Chegaram!

APPLEKIT - Kit de microcomputador tipo Apple®

Componentes para montagem
de um microcomputador
APPLEKIT completo.

microcontrol

Sistemas de Controles
Tels.: (011) 814-0446 e 814-1110
São Paulo - Brasil.



APPLEKIT 65100

APPLEKIT 65020

APPLEKIT 65300

APPLEKIT 65400

APPLEKIT 65000 Placa de circuito impresso. APPLEKIT 65010 Conjunto de soquetes, conectores, resistores e capacitores. APPLEKIT 65020 Conjunto de semicondutores, TTL's, LSI e memórias (As memórias EPROM são fornecidas com gravação). APPLEKIT 65100 Conjunto de teclado alfanumérico com 52 teclas e componentes, circuito impresso. APPLEKIT 65200 Fonte de alimentação tipo chaveado. APPLEKIT 65300 Caixa de microcomputador em poliuretano. APPLEKIT 65400 Manual de montagem e teste de micro.

APPLEKIT é 100% compatível com os cartões periféricos da MICROCRAFT.

* Apple é marca registrada de Apple Inc.

Valmir Benjamin

Além do BASIC, Assembler, DOS e outros mistérios

Roberto Quito de Sant'Anna

Uma das queixas mais frequentes dos usuários do TRSDOS/DOS 500 é a dificuldade na leitura do diretório. Devido à rapidez do rolamento da tela, mesmo utilizando-se a tecla @, fica difícil localizar um programa, principalmente porque, ao contrário do CP/M, os programas não são listados em ordem alfabética. E o que acontece? Bom, se o disco em questão estiver razoavelmente cheio, só conseguiremos achar um programa após várias e irritantes tentativas, e o pior é que, em 99% dos casos, estamos interessados apenas em saber se ele está ou não naquele disco, dispensando todas as demais informações.

A única alternativa disponível, que é chamar o BASIC e apreciar o diretório simplificado CMD"D.d", resolve apenas em parte o problema, pois persiste a necessidade de uma lenta e cuidadosa pesquisa correndo a ponta do dedo indicador sobre as quatro colunas da tela até se achar o programa desejado, uma vez que este diretório simplificado também não é dado em ordem alfabética.

Este problema será a motivação inicial para fazermos neste artigo uma pequena expedição ao território do Assembler e da linguagem de máquina. Aqui será descrita uma sub-rotina em Assembler utilizando rotinas já existentes no DOS para ler ou escrever em qualquer setor de um disco, a qual terá como aplicação dois utilitários: o primeiro fornece um dire-

tório em ordem alfabética do drive desejando (leitura) e o segundo possibilita a mudança do nome e/ou data de um disco (leitura-escrita).

AS FERRAMENTAS DE TRABALHO

As rotinas do TRSDOS/DOS 500 empregadas são:

- CALL 4675H — lê um setor do disco;
- CALL 4600H — escreve um setor no disco.

A primeira rotina copia um setor (256 bytes) do disco para um *buffer* na RAM definido pelo usuário; uma vez feita a cópia, o conteúdo do *buffer* pode ser modificado à vontade e escrito de volta no disco com o auxílio da segunda rotina. Ambas exigem que sejam carregadas nos registradores do Z80 as seguintes informações:

- D — trilha onde se localiza o setor desejado;

- E — setor;
- B — zero;
- C — número do drive (0 — 3);
- HL — endereço inicial do *buffer* de 256 bytes.

A sub-rotina em Assembler, com os respectivos códigos-objeto, está na figura 1. Estes códigos, convertidos para o sistema decimal, serão lidos pelos dois programas e armazenados nas posições BEF0H-BEFCH (48880D-48892D), resultando na linha 80 do primeiro programa ou na 130 do segundo:

DATA 17, 1, 17, 1, 0, 0, 33,
0, 101, 205, 117, 70, 201

Note que o primeiro 1 da sequência (posição BEF1H = 48881D) corresponde ao número do setor e que o primeiro 0 (posição BEF4H = 48884D) corresponde ao número do drive, e que o conteúdo desses endereços pode ser facilmente modificado através de instruções POKE.

11 01 11	LD	DE, 1101H ; trilha em D e setor em E
01 00 00	LD	BC, 0000H ; zero em B e drive em C
21 00 BF	LD	HL, 0BF00H ; end inicial do buffer
CD 75 46	CALL	4675H ; leitura do setor
C9	RET	; volta ao prog BASIC

Figura 1

TRSDOS/DOS 500 X NEWDOS

Para melhor compreensão de nossos aplicativos, torna-se importante, nesta altura, uma breve explanação sobre a organização da trilha 17 do TRSDOS/DOS 500, a qual contém o diretório. É de toda conveniência ter à mão, além do SUPERZAP do NEWDOS, o artigo *O NEWDOS que não está nos manuais*, publicado em MICRO SISTEMAS número 31, uma vez que a organização do diretório nesses dois sistemas apresenta muitos pontos em comum.

Para examinar a trilha 17 do seu disco, coloque no drive 0 o disco com o NEWDOS e tecla SUPERZAP <ENTER>, seguido de DTS<ENTER>; coloque no drive 1 (ou 0, conforme sua configuração) o disco com o TRSDOS/DOS 500 e tecla 1, 17, 1 (ou 0, 17, 1) <ENTER>. Importante: se você não dispõe de NEWDOS, não fique triste: salte por cima da teoria a seguir, digite e usufrua dos aplicativos (e... trate de arranjar um rapidinho!).

Tal como no NEWDOS, o primeiro setor do diretório contém a tabela de alocação de espaço no disco (a partir do byte 00H), o código da senha do disco (bytes CEH e CFH), o nome/data do disco (bytes D0H a DFH), a alocação do comando AUTO a partir de E0H, além de outras informações. Digitando a tecla ;, você verá que o segundo setor também contém os códigos HASH (calculados através do nome/extensão do arquivo — ver opção DNTH do SUPERZAP) correspondentes a cada um dos arquivos, ou o valor 00H se o arquivo foi apagado. Aqui, como no NEWDOS, a sequência dos códigos HASH corresponde exatamente à sequência em que os arquivos aparecem no diretório, o que facilita a identificação.

A partir do terceiro setor (ver figura 2) começam as entradas do diretório propriamente ditas, e aqui reside a principal diferença em relação ao NEWDOS: ao invés de 32 bytes, cada entrada de arquivo utiliza 48 bytes, o que nos dá cinco entradas por setor de 256 bytes e uma sobra de 16 bytes na última linha, bytes FOH a FFH, que são usados pelo

DRV	00	5E07	5200	0042	4153	4943	2020	2043	4D44	..R..BASIC...CMD
0	10	FA17	EF5C	1400	0287	FFFF	FFFF	FFFF	FFFF
0H	20	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF
	30	1000	00FD	0050	4953	444F	4944	4F42	4153PISD0ID0BAS
DRS	40	EF5C	EF5C	0000	1E21	FFFF	FFFF	FFFF	FFFF!
309	50	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF
135H	60	0000	0000	0000	0000	0000	0000	0000	0000
	70	0000	0000	0000	0000	0000	0000	0000	0000
TRK	80	0000	0000	0000	0000	0000	0000	0000	0000
17	90	1000	003E	004C	4F54	4F32	2020	2042	4153>.LOT02...BAS
11H	A0	EF5C	EF5C	0100	0401	FFFF	FFFF	FFFF	FFFF
	B0	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF
TRS	C0	1000	000B	0044	4420	2020	2020	2043	4D44DD.....CMD
3	D0	EF5C	EF5C	0100	0421	FFFF	FFFF	FFFF	FFFF!
3H	E0	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF
	F0	5052	4F4C	4F47	4943	4120	3139	3832	2020	PROLOGICA.1982..

Figura 2 — Display do setor 3 da trilha 17, mostrando quatro entradas de diretório ativas (00H, 30H, 90H e C0H), uma entrada que teve o seu arquivo eliminado por KILL (60H) e a mensagem do fabricante (FOH).

fabricante para vender a sua marca (sabia que com o subcomando MODnn do SUPERZAP você pode colocar aqui qualquer mensagem sua?...). É muito importante notar, como diferença fundamental, que o TRSDOS/DOS 500 zera todos os bytes de uma entrada de diretório cujo arquivo tenha sido apagado por meio de KILL (em vez de zerar apenas o primeiro byte), o que torna muito difícil — mas não impossível — a recuperação de arquivos.

A organização das entradas, a menos dos 48 bytes, é muito semelhante à descrita no artigo sobre o NEWDOS que mencionamos. Atente bem, ainda na figura 2, para o primeiro byte de cada entrada (normalmente 10H para arquivos de usuário não-protetidos e não-invisíveis) e para os bytes de 6 a 16, que contém o nome e extensão de cada arquivo, detalhes a serem explorados por nosso primeiro utilitário. Aconselho, antes de prosseguir na leitura, um passeio de ida e volta ao longo de toda a trilha 17, utilizando as teclas ; para avançar e - para recuar. Note que você sempre poderá saber onde está consultando as informações localizadas à esquerda da tela (figura 2): DRV/drive, DRS/setor relativo no disco, TRK/trilha e TRS/setor relativo na trilha, dados em decimal e em hexadecimal.

Antes de passarmos aos utilitários, uma dica: a linha 40 de ambos os programas realiza a proteção da memória acima de BEEH = 48878D, onde ficarão

armazenados a sub-rotina de leitura-escrita (BEFOH-BEFCH) e o buffer para onde os setores lidos serão transferidos (255 bytes a partir de BF00H). Esta é uma dica muito importante, pois permite-nos executar programas que necessitem de proteção de uma área de memória sem que tenhamos que voltar ao DOS e chamar novamente o BASIC para responder ao Mem. usada? (as posições 16561 e 16562 devem conter, na forma LSB-MSB, o endereço a partir do qual se deseja proteger a memória, sendo indispensável executar logo após um CLEAR n, para que o sistema redefina a localização do espaço de armazenagem string).

UM DIRETÓRIO MAIS PRÁTICO

O programa que está na listagem 1 realiza a listagem em ordem alfabética do diretório de um drive especificado, utilizando a nossa sub-rotina na função de leitura. A essência do programa está contida nas linhas 200-240: um laço controlado pela variável SE percorre os setores de 3 a 18 da trilha 17, transferindo-os, um de cada vez, para o buffer (variável BFS, cujos comprimento e endereço foram modificados na linha 150, colocando-a na área protegida de memória); em seguida, um laço controlado pela variável J percorre as cinco entradas de cada setor e, caso se trate de um arquivo não-protetido e não-invisível (primeiro byte igual a 10H = 16D), trans-

MICRO SISTEMAS Nº 33 . NÃO PERCA!

O número 33 de MICRO SISTEMAS está sensacional! Veja só:
— Com o Editor Assembler para a Linha Sinclair, você terá uma poderosa ferramenta para fazer os seus programas em Assembler. Mais rápido e fácil.

— Apresentando um excelente tratamento de tela, com ilustrações bem interessantes, o programa Cálculo de Volumes permite calcular, automaticamente, o volume de 16 figuras geométricas, a partir de suas dimensões.

— Programação em forma de menu — grande pedida para programadores de todos os níveis. Este artigo mostra a importância das telas interativas e algumas técnicas para a sua elaboração.

— E quem disse que é difícil analisar balanços de empresas? Só se for para quem não tem um TRS-80 e o programa Análise de Balanços.

— Para os usuários do Apple, um sensacional programa — o MILA-GROSSO — que permite recuperar programas aparentemente perdidos após o uso do comando NEW.

fere seu nome/extensão, localizados entre o 6º e 16º bytes em cada entrada, para um elemento da matriz *string* PR\$, a qual, após classificada (CMD "O", linha 250), terá seus elementos impressos à razão de 32 por página de vídeo, sob controle do operador.

PARA IDENTIFICAR SEUS DISCOS

O programa da *listagem 2* demonstra o uso de nossa sub-rotina para ler um setor no disco e escrevê-lo de volta. Após devidamente modificado, possibilita ao usuário alterar os nomes e as datas de seus disquetes com vistas, por exemplo, a uma melhor catalogação, com a vantagem adicional de poder colocar a data em qualquer formato, tal como 20ABR84, ou qualquer outra informação em seu lugar, desde que esta data não ultrapasse oito caracteres.

O programa interroga a quantidade de drives existente no sistema, orientando automaticamente o disquete a ser modificado para o drive 1, se existirem dois ou mais drives (linhas 120-140). O setor lido agora é somente o setor 1 da trilha 17, que contém, entre outras informações, o nome e a data do disquete nos bytes D0H-D7H e D8H-DFH, respectivamente. A leitura propriamente dita e a transferência para o *buffer* são feitas na linha 180, e o nome/data do disquete são armazenados na *string* N\$ (remanejada, na linha 70, para as 16 posições a partir de BFD0H — portanto, nas exatas posições de nome/data do *buffer*), a qual, se desejado, será modificada com os novos nome e/ou data que serão armazenados em N\$ na linha 300.

O detalhe mais interessante a considerar é a modificação da sub-rotina para que ela possa escrever o setor de volta no disco (conforme vimos no início do artigo, as duas rotinas do DOS envolvidas são as localizadas em 4675H (Iê) e 4600H (escreve), cujos parâmetros são exatamente os mesmos e cujos endereços diferem apenas no LSB). Examinando a listagem fonte da sub-rotina, vemos que a instrução CALL 4675H correspondem os códigos-objeto CD 75 46 (atenção: LSB-MSB!), e que o hexadecimal 75 corresponde ao 11º byte da sub-rotina, ou seja, à posição de memória BEFAH, já que a sub-rotina está armazenada a partir de BEFOH. Em consequência, para que a sub-rotina seja transformada em uma de escrita, basta POKEar o decimal 0 na posição BEFAH, conforme a linha 300, chamar a sub-rotina e revertê-la em seguida à condição de leitura, deixando-a em condições de ser utilizada novamente (lembre-se: 117D = 75H).

Detalhes sobre a Listagem 1

- Linha 40 — protege memória a partir de BEEH;
- Linha 120 — leitura e armazenamento da sub-rotina;
- Linha 150 — remanejamento de BFS para o *buffer*;
- Linha 190 — coloca em BEF4H o número do drive;
- Linhas 200 a 240 — corre os setores de 3 a 18; transfere para o *buffer*; seleciona em cada setor arquivos de usuário; transfere nome/ext para matriz PR\$;
- Linha 250 — classifica PR\$ em ordem alfabética;
- Linhas 290 a 350 — lista PR\$ com 32 arquivos por página de vídeo;
- Linhas 380 a 440 — sub-rotinas para as diversas mensagens ao operador.

Listagem 1

```

10 *****LISTA DIRETORIOS EM ORDEM ALFABETICA*****
20 ***Por Roberto QUITO de Sant'Anna - CP 24039 - RJ***
30 *****
40 POKE 16561,&HEE:POKE 16562,&HBE
50 CLEAR 5000:DEFINT A-Z:DIM PR$(90):BFS=""
60 M1$="*****":M2$=STRING$(17,"-")
70 M3$="DIRETORIO EM ORDEM ALFABETICA DO DRIVE ==>"
80 M4$="TECLE <ENTER> PARA NOVA TELA OU <S> PARA SAIR"
90 M5$="<ENTER> PARA NOVO DIRETORIO OU <S> PARA SAIR"
100 M6$="DIRETORIOS EM ORDEM ALFABETICA"
110 M7$=STRING$(63,"*")
120 FOR N=-16656 TO -16644:READ A:POKE N,A:NEXT
130 DATA 17,1,17,1,0,0,33,0,191,205,117,70,201
140 DEFUSR0=&HBEF0:K=0:N=VARPTR(BFS):N1=N+1:N2=N+1
150 POKE N,255:POKE N1,&H0D:POKE N2,&HBF:CLS
160 GOSUB 420 :PRINT@ 530,"QUAL O DRIVE (0-3) > ?";
170 DR$=INKEY$:IF DR$="" THEN 170
180 DR=VAL(DR$):IF DR<0 OR DR>3 THEN CLS:GOTO 160
190 PRINT " ";DR:POKE &HBEF4,DR
200 FOR SE=3 TO 18:POKE &HBEF1,SE:X=USR0(0)
210 FOR J=6 TO 230 STEP 48
220 IF ASC(MID$(BFS,J-5,1))<>16 GOTO 240
230 K=K+1:PR$(K)=MID$(BFS,J,11)
240 NEXT J:NEXT SE
250 CMD "O",K,PR$(1)
260 CLS:PRINT M1$,M3$,DR,M1$:PRINT
270 PRINT M2$,"CONTEM",K,"ARQUIVOS DE USUARIO",M2$
280 POKE 16916,3:CLS:IND=INT(K/32)+1:MSG$=M4$
285 IF K/32=INT(K/32) THEN IND=IND-1
290 FOR I=1 TO IND:FIM=I*32:COM=FIM-31:CLS
300 IF K<FIM THEN FIM=K:MSG$=M5$
310 PRINT:FOR II=COM TO FIM:PRINT PR$(II)+" ";:NEXT
320 IF FIM=K POKE T,61:POKE T+1,61:POKE T+2,62:GOTO 350
330 GOSUB 380 :IF Z$<>"S" GOTO 350
340 CLS:MSG$=M5$:POKE 16916,0:CLS:GOSUB 420 :GOTO 360
350 NEXT I
360 GOSUB 430 :IF Z$="S" POKE 16916,0:CLS:END
370 POKE 16916,0:GOTO 50
380 PRINT@ 905,MSG$:T=15408
390 POKE T,32:POKE T+1,32:POKE T+2,32
400 Z$=INKEY$:IF Z$<>"S" RETURN
410 POKE T,61:POKE T+1,61:POKE T+2,62:GOTO 390
420 PRINT M7$:PRINT TAB(17);M6$:PRINT M7$:RETURN
430 PRINT@ 905,MSG$:Z$=INKEY$:IF Z$="" THEN 430
440 RETURN

```

Bem, espero ter mostrado alguns vestígios da vida inteligente (e como!) que existe além do BASIC. Difícil? Talvez, mas sempre temos a tendência a considerar difícil tudo o que não dominamos e, de qualquer modo, toda dificuldade

encontrada será regamente recompensada pelos novos recursos de que passaremos a dispor: velocidade, flexibilidade e realização pessoal. O que está esperando? Tal como para a maioria das coisas da vida, o difícil é começar...


Detalhes sobre a Listagem 2

- Linha 40 — protege memória a partir de BEEH;
- Linha 50 — leitura e armazenamento da sub-rotina;
- Linha 70 — remanejamento de N1\$ para o buffer;
- Linha 140 — coloca em BEF4H o número do drive;
- Linha 180 — leitura do setor e identificação do nome/data atuais;
- Linhas 210 a 240 — entrada e formatação do novo nome;
- Linhas 250 a 280 — entrada e formatação da nova data;
- Linha 300 — redefinição de N1\$ e mudança da sub-rotina para escrever;
- Linha 310 — escreve o setor e mudança da sub-rotina para ler.

Listagem 2

```
10 *****MUDA NOME/DATA DE DISKETTES TRS DOS/DOS 500*****
20 ***Por Roberto QUITO de Sant' Anna - CP 24039 - RJ***
30 *****
40 POKE 16561,&HEE:POKE 16562,&HBE: CLEAR 200
50 FOR N=-16656 TO -16644:READ A:POKE N,A:NEXT
60 DEFUSR=&HBEF0:N1$="" :N1=VARPTR(N1$)
70 POKE N1,16:POKE N1+1,&HD0:POKE N1+2,&HBF
80 DATA 17,1,17,1,0,0,33,0,191,205,117,70,201
90 M1$="*****":M2$=STRING$(62," ")
100 M3$="MUDA NOME/DATA DISKETTES TRSDOS/DOS 500"
110 CLS:PRINT M1$,M3$:M1$=PRINT M2$:POKE 16916,2
120 PRINT 205,"QUANTOS DRIVES TEM O SEU SISTEMA ";
130 INPUT DR:IF DR=1 THEN DR=0 ELSE DR=1
140 POKE &HBEF4,DR
150 CLS:PRINT 208,"COLOQUE O DISKETTE NO DRIVE";DR
160 PRINT 331,"TECLE <ENTER> PARA MUDAR OU <S> PARA SAIR"
170 Y$=INKEY$:IF Y$="" THEN 170 ELSE IF Y$="S" THEN 320
180 X=USR(0):N2$=LEFT$(N1$,8):D2$=RIGHT$(N1$,8)
190 CLS:PRINT 205,"O NOME ATUAL E' ===> ";D2$
200 PRINT 333,"A DATA ATUAL E' ===> ";D2$
210 PRINT 512,"ENTRE NOVO NOME OU <ENTER> PARA MANTER ";
220 LINE INPUT N$:IF N$="" N$=N2$:GOTO 250
230 IF LEN(N$)<8 THEN N$=N$+STRING$(8-LEN(N$)," ")
240 IF LEN(N$)>8 THEN N$=LEFT$(N$,8)
250 PRINT 640,"ENTRE NOVA DATA OU <ENTER> PARA MANTER ";
260 LINE INPUT D$:IF D$="" THEN D$=D2$:GOTO 290
270 IF LEN(D$)<8 THEN D$=D$+STRING$(8-LEN(D$)," ")
280 IF LEN(D$)>8 THEN D$=LEFT$(D$,8)
290 CLS:PRINT 592,"GRAVANDO ";N$;"(---)";D$
300 N$=N$+D$:MID$(N1$,1,16)=N$:POKE &HBEFA,0
310 X=USR(0):POKE &HBEFA,117:FOR N=1 TO 1000:NEXT
320 CLS:PRINT 843,"";
330 PRINT "<ENTER> PARA NOVO DISKETTE OU <S> PARA SAIR"
340 Z$=INKEY$:IF Z$="" THEN 340
350 IF Z$="S" POKE 16916,0:CLS:END ELSE GOTO 150
```

Roberto Quito de Sant'Anna é engenheiro de telecomunicações, formado pelo Instituto Militar de Engenharia. Professor da cadeira de Informática da Academia Militar das Agulhas Negras, desde agosto de 1982 é também colaborador da MICRO-MAXI Computadores e Sistemas, como analista de sistemas.




EDISA
 A solução lógica

TODA A LINHA DE MICROS E MINICOMPUTADORES

A informação nas pontas dos seus dedos


Gerenciador de banco de dados



Sistemas operacionais:
MS-DOS (IBM-PC) CP/M
M CP/M-86 CROMIX
CDOS MP/M MP/M-86

☐ Cursor endereçável para o uso de funções fullscreen.

Automação de escritórios



Requerimentos do Sistema:

☐ (Microprocessadores 8080, 8085, Z80, 8086, 8088)

☐ 64K p/ CP/M; 128K p/ CP/M-86 e MS-DOS; 56K p/ Apple II

☐ 2 Disketes c/ mínimo de 126K cada

☐ Terminal c/80 colunas e cursor endereçável

☐ Impressora com no mínimo 80 colunas.

Diamac (série 8.100) Microdigital (TK's 83/85/2000) CP's 200/300/500, impressoras, eibras, eigh, diamac Microengenho I e II e Apple-Tronic

Suprimentos
 Discos Magnéticos: RMS, 16MB, 40MB, etc.
 Disquetes: 5 1/4" e 3 1/2" singles e dupla faces
 Etiquetas (várias marcas)
 Fitas Magnéticas: 900, 1200 e 2400 Pps
 Fitas p/impressoras: eibras, digitab, diablo, centronic, etc.
 Cartuchos Cobra 400
 Pastas e formulários contínuos

End. Rua da Lapa, 180 gr.1108 a 1110 - CEP 20021 - Rio de Janeiro - Tel.: (021) 221-3069

MICRO PROCESS COMPUTADORES LTDA.

- CENTRO DE MICROS**
- AVALIAÇÃO DE CARGA
 - PLANEJAMENTO
 - DIMENSIONAMENTO
 - IMPLANTAÇÃO DE EQUIPAMENTOS
 - PROGRAMAS
 - TREINAMENTO

Implantação racionalizada de micro-centros para

- Comércio
- Indústria
- Profis. Liberais

EQUIPAMENTOS

- Microdigital
- Prológica
- Unitron. CCE. etc..

SOFTWARE

- Programas especiais personaliz. de softer
 - Pacotes p/ advogados
- Despachamos Via Varig

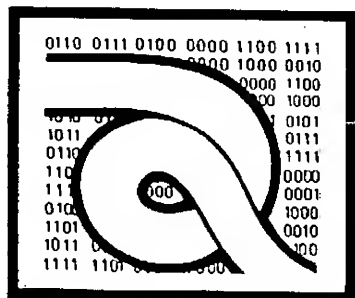
AMPLO FINANCIAMENTO

TEL.: 64-0468

Alameda Lorena, nº 1310

CEP 01424 - São Paulo

ESTACIONAMENTO PARA CLIENTES



Curso de Assembler – XVI

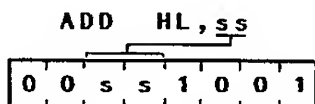
Como já pudemos observar durante o curso, o microprocessador Z80 tem a capacidade de realizar operações com 16 bits. Nesta lição, vamos conhecer o *grupo de aritmética de 16 bits*, o qual se refere às operações aritméticas que podem ser realizadas com pares de registradores (por exemplo, HL). Vamos conhecer estas instruções.

1 – Adição de registradores de 16 bits

Formato: ADD HL, ss

Operação: Efetua uma operação de adição entre o conteúdo do par de registradores HL e outro par de registradores de 16 bits.

Código Objeto:

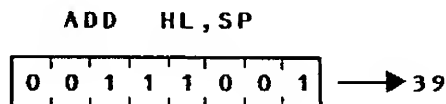


(para o par ss, veja figura 1)

ss = 00	para o par de registradores BC
ss = 01	para o par de registradores DE
ss = 10	para o par de registradores HL
ss = 11	para o par de registradores SP

Figura 1

Exemplo:



Descrição: O conteúdo do par de registradores ss (qualquer dos pares de registradores: BC, DE, HL ou SP) é somado ao conteúdo do par de registradores HL e o resultado é guardado em HL.

ADD HL, ss HL ← HL + ss

Ciclos de máquina (M) : 3

States (T): 11(4,4,3)

Flags afetadas: S – não afetada;
Z – não afetada;
H – setada se ocorre *vai-um* do bit 11; senão é ressetada;
P/V – não afetada;
N – ressetada;
C – setada se ocorre *vai-um* do bit 15; senão é ressetada.

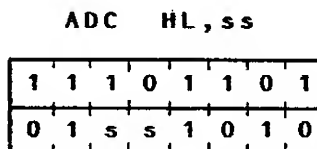
Um exemplo: se o par de registradores HL contém o inteiro 4242H e o par de registradores DE contém 1111H, após a execução de ADD HL, DE, o conteúdo do par de registradores HL será 5353H.

2 – Adição de registradores de 16 bits com Carry

Formato: ADC HL, ss

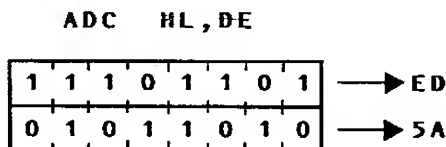
Operação: Efetua uma operação de adição entre: o conteúdo do par de registradores HL, o outro par de registradores de 16 bits e o conteúdo da *flag Carry*.

Código Objeto:



(para o par ss, veja figura 1)

Exemplo:



Descrição: O conteúdo do par de registradores ss (qualquer dos pares de registradores: BC, DE, HL ou SP) é somado com a *flag Carry* e com o conteúdo do par de registradores HL; o resultado é guardado no par de registradores HL.

ADC HL, ss $HL \leftarrow HL + ss + CY$

Ciclos de máquina (M): 4

States (T): 15(4,4,3)

Flags afetadas: S - setada se o resultado é negativo; senão é ressetada;
Z - setada se o resultado é zero; senão é ressetada;
H - setada se ocorre *vai-um* do bit 11; senão é ressetada;
P/V - setada se ocorre um *overflow*; senão é ressetada;
N - ressetada;
C - setada se ocorre *vai-um* do bit 15; senão é ressetada.

Um exemplo: se o par de registradores BC contém 2222H, o par de registradores HL contém 5437H e o *Carry* está setado; após a execução de ADC HL, BC, o conteúdo de HL será 765A H.

3 - Subtração de registradores de 16 bits com Carry

Formato: SBC HL, ss

Operação: Efetua uma operação de subtração entre: o conteúdo do par de registradores HL, o outro par de registradores de 16 bits e o conteúdo da *flag Carry*.

Código Objeto:

SBC HL, ss

1	1	1	0	1	1	0	1
0	1	s	s	0	0	1	0

(para o par ss, veja figura 1)

Exemplo:

SBC HL, DE

1	1	1	0	1	1	0	1
0	1	0	1	0	0	1	0

→ ED

→ 52

Descrição: O conteúdo do par de registradores ss (qualquer dos pares de registradores: BC, DE, HL ou SP) e a *flag Carry* são subtraídos do conteúdo do par de registradores HL, e o resultado é guardado no par de registradores HL.

SBC HL, ss $HL \leftarrow HL - ss - CY$

Ciclos de máquina (M): 4

States (T): 15(4,4,3)

Flags afetadas: S - setada se o resultado é negativo; senão é ressetada;
Z - setada se o resultado é zero; senão é ressetada;
H - setada se ocorre um empréstimo do bit 12; senão é ressetada;
P/V - setada se ocorre um *overflow*; senão é ressetada;
N - setada;
C - setada se ocorre um empréstimo; senão é ressetada.

Um exemplo: se o conteúdo do par de registradores HL é 9999H, o conteúdo do par de registradores DE é 1111H e a *flag Carry* está setada; após a execução de SBC HL, DE, o conteúdo de HL será 8887H.

4 - Adição de registrador indexador com par de registradores

Formato: ADD IX, pp

Operação: Efetua uma operação de adição entre o conteúdo do registrador indexador (IX ou IY) com um par de registradores de 16 bits.

Código Objeto:

ADD IX, pp

1	1	0	1	1	1	0	1
0	0	r	r	1	0	0	1

→ DD

ADD IY, rr

1	1	1	1	1	1	0	1
0	0	r	r	1	0	0	1

→ FD

CÓDIGO	pp	rr
00	BC	BC
01	DE	DE
10	IX	IY
11	SP	SP

Exemplo:

ADD IX, BC

1	1	0	1	1	1	0	1
0	0	0	0	1	0	0	1

→ DD

→ 09

Descrição:

O conteúdo do par de registradores pp ou rr (qualquer dos registradores: BC, DE, SP ou IX/IY) é somado ao conteúdo do par de registradores IX ou IY, e o resultado é guardado no par de registrador indexador.

**Micro
Sistemas**

**GARANTA
SUA MS
TODO MÊS!**

Se você deseja assinar MICRO SISTEMAS, preencha o cupom abaixo (ou uma xerox, caso você não queira cortar a revista):

nome _____

empresa _____

profissão/cargo _____

endereço para remessa _____

cidade _____ cep _____ estado _____

Assinatura anual

☐ Micro Sistemas Cr\$ 20.000,00

☐ Informática & Administração Cr\$ 20.000,00

☐ Micro Sistemas + Informática & Administração Cr\$ 36.000,00

Preencha um cheque nominal à ATI Editora Ltda e envie para:

Av. Presidente Wilson, 165/grupo 1210, Centro, Rio de Janeiro, RJ, CEP 20030 - tels.: (021) 262-5259 e 262-5208.

R. Oliveira Dias, 153, Jardim Paulista, São Paulo, SP, CEP 01433 - tels.: (011) 853-7758, 881-5668 e 853-3800.

Seu recibo será enviado pelo correio.

ADD IX, pp $IX \leftarrow IX + pp$

ADD IY, rr $IY \leftarrow IY + rr$

Ciclos de máquina (M): 4

States (T): 15(4,4,4,3)

Flags afetadas: S - não afetada;
Z - não afetada;
H - setada se ocorre *vai-um* do bit 11; senão é ressetada;
P/V - não afetada;
N - ressetada;
C - setada se ocorre *vai-um* do bit 15; senão é ressetada.

Um exemplo: se o conteúdo do registrador indexador IY é 333H e o conteúdo do par de registradores BC é 555H, após a execução de ADD IY, BC, o conteúdo de IY será 888H.

5 - Incrementa par de registradores

Formato: INC ss

Operação: Incrementa o conteúdo do par de registradores especificado.

Código Objeto:

INC ss

0	0	s	s	0	0	1	1
---	---	---	---	---	---	---	---

(para o par ss, veja figura 1)

Exemplo:

INC HL

0	0	1	0	0	0	1	1
---	---	---	---	---	---	---	---

→ 23

Descrição: O conteúdo do par de registradores ss (qualquer dos pares de registradores: BC, DE, HL ou SP) é incrementado.

INC ss $ss \leftarrow ss + 1$

Ciclos de máquina (M): 1

Flags afetadas: Nenhuma.

Um exemplo: se o par de registradores HL contém 1000H, após a execução de INC HL, o par de registradores HL conterá 1001H.

6 - Incrementa registrador indexador

Formato: INC IX

INC IY

Operação: Incrementa o conteúdo do registrador indexador IX ou IY.

Código Objeto:

INC IX

1	1	0	1	1	1	0	1
0	0	1	0	0	0	1	1

→ DD
→ 23

INC IY

1	1	1	1	1	1	0	1
0	0	1	0	0	0	1	1

→ FD
→ 23

Descrição: O conteúdo do registrador IX ou IY é incrementado.

INC IX $IX \leftarrow IX + 1$

INC IY $IY \leftarrow IY + 1$

Ciclos de máquina (M): 2

States (T): 10(4,6)

Flags afetadas: Nenhuma.

Um exemplo: se o conteúdo do registrador IX é 3300H, após a execução de INC IX, o conteúdo do registrador IX será 3301H.

7 - Decrementa par de registradores

Formato: DEC ss

Operação: Decrementa o conteúdo do par de registradores especificado.

Código Objeto:

DEC ss

0	0	s	s	1	0	1	1
---	---	---	---	---	---	---	---

(para o par ss, veja figura 1)

Exemplo:

DEC HL

0	0	1	0	1	0	1	1
---	---	---	---	---	---	---	---

→ 1B

Descrição: O conteúdo do par de registradores ss (qualquer dos pares de registradores: BC, DE, HL ou SP) é decrementado.

DEC ss $ss \leftarrow ss - 1$

Ciclos de máquina (M): 1

States (T): 6

Flags afetadas: Nenhuma.

Um exemplo: se o conteúdo do par de registradores HL é 1001H, após a execução de DEC HL, o conteúdo de HL será 1000H.

8 - Decrementa registrador indexador

Formato: DEC IX

DEC IY

Operação: Decrementa o conteúdo do registrador indexador IX ou IY.

Código Objeto:

DEC IX

1	1	0	1	1	1	0	1
0	0	1	0	1	0	1	1

→ DD
→ 2B

DEC IY

1	1	1	1	1	1	0	1
0	0	1	0	1	0	1	1

→ FD
→ 2B

Descrição: O conteúdo do registrador IX ou IY é decrementado.

DEC IX $IX \leftarrow IX - 1$

DEC IY $IY \leftarrow IY - 1$

Ciclos de máquina (M): 2

States (T): 10(4,6)

Flags afetadas: Nenhuma.

Um exemplo: se o conteúdo do registrador IY é 7649H, após a execução de DEC IY, o conteúdo do registrador IY será 7648H.

Estamos chegando ao fim das instruções do microprocessador Z80. Após completarmos a descrição de todas as instruções disponíveis, serão divulgadas algumas dicas para a elaboração de programas em Assembler. Daí para frente, o resto é com você. Até a próxima lição.

Amaury Correa de Almeida Moraes Junior é formado pelo curso de Análise de Sistemas da FASP, tendo feito diversos cursos de aperfeiçoamento nas áreas de Eletrônica Digital e Microprocessadores, e atualmente trabalha na área de microcomputadores para o Citybank.

NOVO CP 300 PROLÓGICA.

O pequeno grande micro.

Agora, na hora de escolher entre um microcomputador pessoal simples, de fácil manejo e um sofisticado microcomputador profissional, você pode ficar com os dois.

Porque chegou o novo CP 300 Prológica.

O novo CP 300 tem preço de microcomputador pequeno. Mas memória de microcomputador grande.

Ele já nasceu com 64 kbytes de memória interna com possibilidade de expansão de memória externa para até quase 1 megabyte.

E tem um teclado profissional, que dá ao CP 300 uma versatilidade incrível. Ele pode ser utilizado com programas de fita cassete, da mesma maneira que com programas em disco.

64K

O único na sua faixa que já nasce com 64 kbytes de memória.



Compatível com programas em fita cassete ou em disco.

Pode ser ligado ao seu aparelho de TV, da mesma forma que no terminal de vídeo de uma grande empresa.

Com o CP 300 você pode fazer conexões telefônicas para coleta de dados,

se utilizar de uma impressora e ainda dispor de todos os programas existentes para o CP 500 ou o TRS-80 americano. E o que é melhor:

você estará apto a operar qualquer outro sistema de microcomputador.

Nenhum outro microcomputador pessoal na sua faixa tem tantas possibilidades de expansão ou desempenho igual.

CP 300 Prológica.

Os outros não fazem o que ele faz, pelo preço que ele cobra.



PROLOGICA
microcomputadores

Av. Eng.º Luis Carlos Berrini, 1168 - SP



AM
Manaus - 234-1045
• BA-Salvador - 247-8951
• CE-Fortaleza - 226-1523 - 226-0871 - 244-2448
• DF-Brasília - 226-1523 - 226-4534 • ES-Vila Velha
229-1387 - Vitória - 222-5811 • GO-Goiânia - 224-7098 • MT
Cuiabá - 321-2307 • MS-Campo Grande - 383-1270 - Dourados - 421-1052
• MG-Belo Horizonte - 227-0881 - Belim - 531-3806 - Cel. Fabriciano - 841-3400 - Juiz
de Fora - 212-9075 - Uberlândia - 235-1099 • PA-Belém - 228-0011 • PR-Cascavel - 23-1538 - Curitiba - 224-5616 - 224-3422 - Foz do Iguaçu - 73-3734 - Londrina - 23-0065 • PE-Recife - 221-0142 • PI-Teresina
222-0186 • RJ-Campos - 22-3714 - Rio de Janeiro - 264-5797 - 253-3395 - 252-2050 • RN-Natal - 222-3212 • RS-Caxias do
Sul - 221-3516 - Pelotas - 22-9918 - Porto Alegre - 22-4800 - 24-0311 - Santa Rosa - 512-1399 • RO-Porto Velho - 221-2656 • SP
Barretos - 22-6411 - Campinas - 2-4483 - Jundiaí - 434-0222 - Marília - 33-5099 - Mogi das Cruzes - 469-6640 - Piracicaba - 33-1470 - Ribeirão
Preto - 625-5926 - 635-1195 - São Joaquim da Barra - 728-2472 - São José dos Campos - 22-7311 - 22-4740 - São José do Rio Preto - 32-2842 - Santos - 33-2230
Sorocaba - 33-7794 • SC-Blumenau - 22-6277 - Chapecó - 22-0001 - Criciúma - 33-2604 - Florianópolis - 22-9622 - Joinville - 33-7520 • SE-Aracaju - 224-1310.

Solicite demonstração nos principais magazines.



A Microdigital apresenta o novo TK 2000 color.

Que tal um micro de alta performance, que traz as principais características dos equipamentos mais sofisticados e que não exige de você um grande investimento inicial? E que tal um micro que cresce de acordo com as suas necessidades? São estas as vantagens que vão fazer do novíssimo TK 2000 Color um dos maiores sucessos no setor.

Veja: ele tem 64K de memória RAM e 16K de memória ROM, teclado profissional tipo máquina de escrever, recebe diskette e impressora com interface já contido, trabalha em cores, oferece alta resolução gráfica e som.

Peça uma demonstração. Nunca tanto foi lhe oferecido por tão pouco. Grande quantidade de software disponível (entre eles: diversos aplicativos comerciais e jogos a cores de alta resolução gráfica).

Ele tem tudo que um micro deve ter. Menos o preço.

MICRODIGITAL

Caixa Postal 54088 - CEP 01000 - São Paulo - SP - Telex Nº (011) 37.008 - Mide BR
À venda nas boas casas do ramo, lojas especializadas de fotovídeo-som e grandes magazines.
Se você não encontrar este equipamento na sua cidade ligue para (011) 800-255-8583.